目　录

第 1 章　小波分析的基本理论

小波分析属于时频分析的一种。传统的信号分析是建立在傅里叶(Fourier)变换的基础之上的，由于傅里叶分析使用的是一种全局的变换，要么完全在时域，要么完全在频域，因此无法表述信号的时频局域性质，而这种性质恰恰是非平稳信号最根本和最关键的性质。为了分析和处理非平稳信号，人们对傅里叶分析进行了推广乃至根本性的革命，提出并发展了一系列新的信号分析理论：短时傅里叶变换、Gabor 变换、时频分析、小波变换、Randon－Wigner 变换、分数阶傅里叶变换、线调频小波变换、循环统计量理论和调幅－调频信号分析等。其中，短时傅里叶变换和小波变换也是应传统的傅里叶变换不能够满足信号处理的要求而产生的。短时傅里叶变换分析的基本思想是：假定非平稳信号在分析窗函数 $g(t)$ 的一个短时间间隔内是平稳(伪平稳)的，并移动分析窗函数，使 $f(t)g(t-\tau)$ 在不同的有限时间宽度内是平稳信号，从而计算出各个不同时刻的功率谱。但从本质上讲，短时傅里叶变换是一种单一分辨率的信号分析方法，因为它使用一个固定的短时窗函数。因而短时傅里叶变换在信号分析上还是存在着不可逾越的缺陷。

小波变换是一种信号的时间—尺度(时间—频率)分析方法，它具有多分辨率分析(Multiresolution Analysis)的特点，而且在时频两域都具有表征信号局部特征的能力，是一种窗口大小固定不变但其形状可改变，时间窗和频率窗都可以改变的时频局部化分析方法。即在低频部分具有较高的频率分辨率和较低的时间分辨率，在高频部分具有较高的时间分辨率和较低的频率分辨率，很适合于探测正常信号中夹带的瞬态反常现象并展示其成分，所以被誉为分析信号的显微镜，利用连续小波变换进行动态系统故障检测与诊断具有良好的效果。

1.1　傅里叶变换到小波分析

1.1.1　傅里叶变换

傅里叶变换是众多科学领域(特别是信号处理、图像处理、量子物理等)里重要的应用工具之一。从实用的观点看，当人们考虑傅里叶分析的时候，通常是指(积分)傅里叶变换和傅里叶级数。

定义 1.1　函数 $f(t)\in L^1(R)$ 的连续傅里叶变换定义为

$$F(\bar{\omega}) = \int_{-\infty}^{\infty} e^{-i\omega t} f(t)\mathrm{d}t \tag{1.1}$$

$F(\bar{\omega})$ 的傅里叶逆变换定义为

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{i\omega t} F(\omega)\mathrm{d}\omega \tag{1.2}$$

为了计算傅里叶变换，需要用数值积分，即取 $f(t)$ 在 R 上的离散点上的值来计算这个

积分。在实际应用中，我们希望在计算机上实现信号的频谱分析及其它方面的处理工作，对信号的要求是：在时域和频域应是离散的，且都是有限长。下面我们将给出离散时间傅里叶变换(Discrete Fourier Transform，简称DFT)的定义。

定义 1.2　给定实的或复的离散时间序列 f_0，f_1，…，f_{N-1}，设该序列绝对可和，即满足 $\sum_{n=0}^{N-1}|f_n|<\infty$，称

$$X(k)=F(f_n)=\sum_{n=0}^{N-1}f_n e^{-i\frac{2\pi k}{N}n}\qquad k=0,1,\cdots,N-1 \tag{1.3}$$

为序列$\{f_n\}$的离散傅里叶变换，称

$$f_n=\frac{1}{N}\sum_{k=0}^{N-1}X(k)e^{i\frac{2\pi k}{N}n}\qquad n=0,1,\cdots,N-1 \tag{1.4}$$

为序列$\{X(k)\}$的逆离散傅里叶变换(IDFT)。

式(1.4)中，n 相当于对时间域的离散化，k 相当于频率域的离散化，且它们都是以 N 点为周期的。离散傅里叶变换序列$\{X(k)\}$是以 2π 为周期的，且具有共轭对称性。

当 $f(t)$是实轴上以 2π 为周期的函数时，即 $f(t)\in L^2(0,2\pi)$，则 $f(t)$可以表示成傅里叶级数的形式，即

$$f(t)=\sum_{n=-\infty}^{\infty}C_n e^{-int} \tag{1.5}$$

傅里叶变换是时域到频域互相转化的工具，从物理意义上讲，傅里叶变换的实质是把 $f(t)$这个波形分解成许多不同频率的正弦波的叠加和。这样我们就可以把对原函数 $f(t)$的研究转化为对其权系数，即其傅里叶变换 $F(\omega)$的研究。从傅里叶变换中可以看出，这些标准基是由正弦波及其高次谐波组成的，因此它在频域内是局部化的。

例 1　在某工程实际应用中，有一信号的主要频率成分是 50 Hz 和 300 Hz 的正弦信号，该信号被一白噪声污染，现对该信号进行采样，采样频率为 1000 Hz，通过傅里叶变换对其频率成分进行分析。

解：该问题实质是利用傅里叶变换对信号进行频域分析，其 MATLAB 程序如下：

```
t=0:0.001:1.3；%时间间隔为 0.001 说明采样频率为 1000 Hz
x=sin(2*pi*50*t)+sin(2*pi*300*t)；%产生主要频率为 50 Hz 和 300 Hz 的信号
f=x+3.5*randn(1,length(t))；%在信号中加入白噪声
subplot(321)；plot(f)；%画出原始信号的波形图
Ylabel('幅值')；
Xlabel('时间')；
title('原始信号')；
y=fft(f,1024)；%对原始信号进行离散傅里叶变换,参加 DFT 的采样点个数为 1024
p=y.*conj(y)/1024；%计算功率谱密度
ff=1000*(0:511)/1024；%计算变换后不同点所对应的频率值
subplot(322)；plot(ff,p(1:512))；%画出信号的频谱图
Ylabel('功率谱密度')；
```

```
Xlabel('频率');
title('信号功率谱图');
```

输出结果如图 1.1 所示：

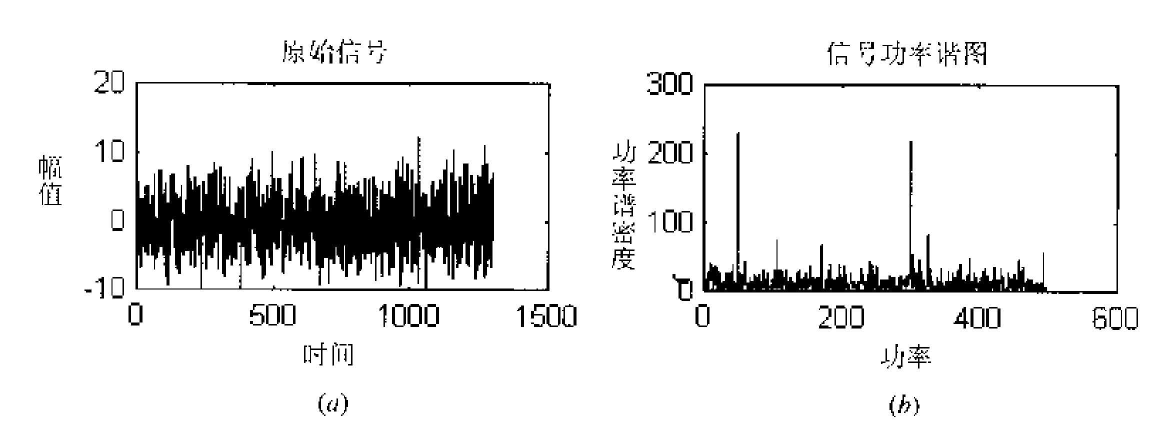


图 1.1　输出结果

从图 1.1(a)中我们看不出任何频域的性质。但从信号的功率谱图(图 1.1(b))中，我们可以明显看出该信号是由频率为 50 Hz 和 300 Hz 的正弦信号和频率分布广泛的白噪声信号组成的，可以明显地看出信号的频率特性。

虽然傅里叶变换能够将信号的时域特征和频域特征联系起来，能分别从信号的时域和频域观察，但却不能把二者有机地结合起来。这是因为信号的时域波形中不包含任何频域信息。而其傅里叶谱是信号的统计特性，从其表达式中也可以看出，它是整个时间域内的积分，没有局部化分析信号的功能，完全不具备时域信息，也就是说，对于傅里叶谱中的某一频率，不知道这个频率是在什么时候产生的。这样在信号分析中就面临一对最基本的矛盾：时域和频域的局部化矛盾。

在实际的信号处理过程中，尤其是对非平稳信号的处理中，信号在任一时刻附近的频域特征都很重要。如柴油机缸盖表面的振动信号就是由撞击或冲击产生的，是一瞬变信号，仅从时域或频域上来分析是不够的。这就促使去寻找一种新方法，能将时域和频域结合起来描述观察信号的时频联合特征，构成信号的时频谱。这就是所谓的时频分析法，亦称为时频局部化方法。

1.1.2　短时傅里叶变换

由于标准傅里叶变换只在频域里有局部分析的能力，而在时域里不存在这种能力，Dennis Gabor 于 1946 年引入了短时傅里叶变换(Short-time Fourier Transform)。短时傅里叶变换的基本思想是：把信号划分成许多小的时间间隔，用傅里叶变换分析每一个时间间隔，以便确定该时间间隔存在的频率。其表达式为

$$S(\omega,\ \tau) = \int_R f(t)g^*(\overline{\omega} - \tau)\mathrm{e}^{-\mathrm{i}\omega t}\mathrm{d}t \tag{1.6}$$

其中“ * ”表示复共轭，$g(t)$是有紧支集的函数，$f(t)$是进入分析的信号。在这个变换中，$\mathrm{e}^{\mathrm{i}\omega t}$起着频限的作用，$g(t)$起着时限的作用。随着时间 τ 的变化，$g(t)$所确定的“时间窗”在 t 轴上移动，使 $f(t)$“逐渐”进行分析。因此，$g(t)$往往被称之为窗口函数，$S(\omega,\tau)$大致反映了 $f(t)$在时刻 τ 时、频率为 ω 的“信号成分”的相对含量。这样信号在窗函数上的展开就可以

表示为在$[\tau-\delta, \tau+\delta]$、$[\omega-\varepsilon, \omega+\varepsilon]$这一区域内的状态，并把这一区域称为窗口，$\delta$和$\varepsilon$分别称为窗口的时宽和频宽，表示了时频分析中的分辨率，窗宽越小则分辨率就越高。很显然，希望δ和ε都非常小，以便有更好的时频分析效果，但海森堡(Heisenberg)测不准原理(Uncertainty Principle)指出δ和ε是互相制约的，两者不可能同时都任意小(事实上，$\delta\varepsilon \geqslant \frac{1}{2}$，且仅当$g(t)=\frac{1}{\delta\pi^{1/4}}e^{-\frac{t^2}{2\delta^2}}$为高斯函数时，等号成立)，如图1.2所示。

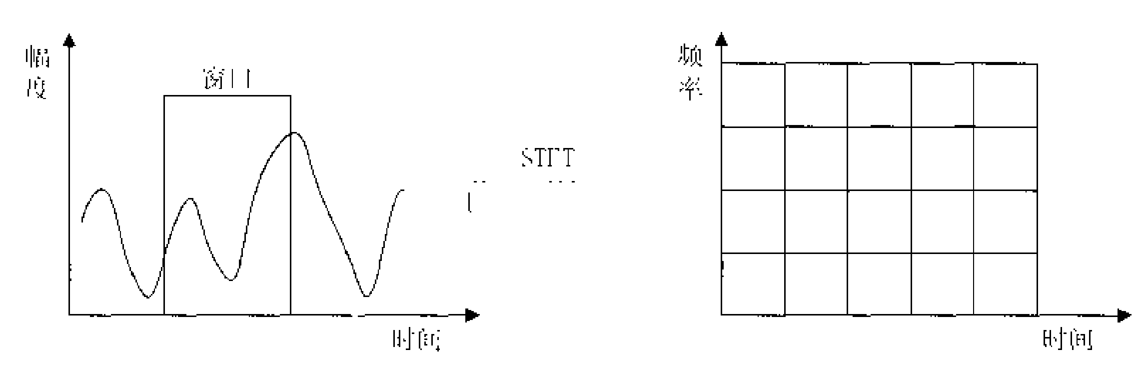


图 1.2

由此可见，短时傅里叶变换虽然在一定程度上克服了标准傅里叶变换不具有局部分析能力的缺陷，但它也存在着自身不可克服的缺陷，即当窗函数$g(t)$确定后，矩形窗口的形状就确定了，τ，ω只能改变窗口在相平面上的位置，而不能改变窗口的形状。可以说短时傅里叶变换实质上是具有单一分辨率的分析，若要改变分辨率，则必须重新选择窗函数$g(t)$。因此，短时傅里叶变换用来分析平稳信号犹可，但对非平稳信号，在信号波形变化剧烈的时刻，主频是高频，要求有较高的时间分辨率(即δ要小)，而波形变化比较平缓的时刻，主频是低频，则要求有较高的频率分辨率(即ε要小)。而短时傅里叶变换不能兼顾两者。

1.1.3 小波分析

小波分析方法是一种窗口大小(即窗口面积)固定但其形状可改变，时间窗和频率窗都可改变的时频局部化分析方法。即在低频部分具有较高的频率分辨率和较低的时间分辨率，在高频部分具有较高的时间分辨率和较低的频率分辨率，所以被誉为数学显微镜。正是这种特性，使小波变换具有对信号的自适应性。

小波分析是调和分析这一数学领域半个世纪以来的工作结晶，已经和必将广泛地应用于信号处理、图像处理、量子场论、地震勘探、语音识别与合成、音乐、雷达、CT成像、彩色复印、流体湍流、天体识别、机器视觉、机械故障诊断与监控、分形以及数字电视等科技领域。原则上讲，传统上使用傅里叶分析的地方，都可以用小波分析取代。小波分析优于傅里叶变换的地方是，它在时域和频域同时具有良好的局部化性质。

设$\Psi(t)\in L^2(R)$($L^2(R)$表示平方可积的实数空间，即能量有限的信号空间)，其傅里叶变换为$\hat{\Psi}(\omega)$，当$\hat{\Psi}(\omega)$满足允许条件(Admissible Condition)：

$$C_\Psi = \int_R \frac{|\hat{\Psi}(\omega)|^2}{|\omega|} d\omega < \infty \tag{1.7}$$

时，我们称$\Psi(t)$为一个基本小波或母小波(Mother Wavelet)。将母函数$\Psi(t)$经伸缩和平移

后，就可以得到一个小波序列。

对于连续的情况，小波序列为

$$\Psi_{a,b}(t) = \frac{1}{\sqrt{|a|}}\Psi\left(\frac{t-b}{a}\right) \qquad a,b \in R;\ a \neq 0 \tag{1.8}$$

其中 a 为伸缩因子，b 为平移因子。

对于离散的情况，小波序列为

$$\Psi_{j,k}(t) = 2^{-j/2}\Psi(2^{-j}t - k) \qquad j,k \in Z$$

对于任意的函数 $f(t) \in L^2(R)$ 的连续小波变换为

$$W_f(a,b) = \langle f, \Psi_{a,b}\rangle = |a|^{-1/2}\int_R f(t)\,\overline{\Psi\left(\frac{t-b}{a}\right)}\mathrm{d}t \tag{1.9}$$

其逆变换为

$$f(t) = \frac{1}{C_\Psi}\int_R\int_R \frac{1}{a^2} W_f(a,b)\Psi\left(\frac{t-b}{a}\right)\mathrm{d}a\mathrm{d}b \tag{1.10}$$

小波变换的时频窗口特性与短时傅里叶的时频窗口不一样。其窗口形状为两个矩形 $[b-a\Delta\Psi,\ b-a\Delta\Psi]\times[(\pm\omega_0-\Delta\hat{\Psi})/a,\ (\pm\omega_0,\ +\Delta\hat{\Psi})/a]$，窗口中心为 $(b,\ \pm\omega_0/a)$，时窗和频窗宽分别为 $a\Delta\Psi$ 和 $\Delta\hat{\Psi}/a$。其中 b 仅仅影响窗口在相平面时间轴上的位置，而 a 不仅影响窗口在频率轴上的位置，也影响窗口的形状。这样小波变换对不同的频率在时域上的取样步长是调节性的，即在低频时小波变换的时间分辨率较差，而频率分辨率较高；在高频时小波变换的时间分辨率较高，而频率分辨率较低，这正符合低频信号变化缓慢而高频信号变化迅速的特点。这便是它优于经典的傅里叶变换与短时傅里叶变换的地方。从总体上来说，小波变换比短时傅里叶变换具有更好的时频窗口特性。

1.1.4 小波分析与傅里叶变换的比较

小波分析是傅里叶分析思想方法的发展与延拓。它自产生以来，就一直与傅里叶分析密切相关。它的存在性证明，小波基的构造以及结果分析都依赖于傅里叶分析，二者是相辅相成的。两者相比较主要有以下不同：

(1) 傅里叶变换的实质是把能量有限信号 $f(t)$ 分解到以 $\{e^{i\omega t}\}$ 为正交基的空间上去；小波变换的实质是把能量有限信号 $f(t)$ 分解到 $W_{-j}(j=1,2,\cdots,J)$ 和 V_{-J} 所构成的空间上去。

(2)傅里叶变换用到的基本函数只有 $\sin(\bar{\omega}t)$，$\cos(\bar{\omega}t)$，$\exp(i\bar{\omega}t)$，具有唯一性；小波分析用到的函数(即小波函数)则具有不唯一性，同一个工程问题用不同的小波函数进行分析有时结果相差甚远。小波函数的选用是小波分析应用到实际中的一个难点问题(也是小波分析研究的一个热点问题)，目前往往是通过经验或不断的试验(对结果进行对照分析)来选择小波函数。

(3) 在频域中，傅里叶变换具有较好的局部化能力，特别是对于那些频率成分比较简单的确定性信号，傅里叶变换很容易把信号表示成各频率成分的叠加和的形式。例如，$\sin(\bar{\omega}_1 t) - 0.345\sin(\bar{\omega}_2 t) + 4.23\cos(\bar{\omega}_3 t)$，但在时域中，傅里叶变换没有局部化能力，即无法从信号 $f(t)$ 的傅里叶变换 $\hat{f}(\omega)$ 中看出 $f(t)$ 在任一时间点附近的性态。事实上，$\hat{f}(\bar{\omega})\mathrm{d}\bar{\omega}$ 是关于频率为 $\bar{\omega}$ 的谐波分量的振幅，在傅里叶展开式中，它是由 $f(t)$ 的整体性态所决定的。

(4) 在小波分析中，尺度a的值越大相当于傅里叶变换中$\bar{\omega}$的值越小。

(5) 在短时傅里叶变换中，变换系数$S(\bar{\omega},\tau)$主要依赖于信号在$[\tau-\delta, \tau+\delta]$片段中的情况，时间宽度是$2\delta$(因为$\delta$是由窗函数$g(t)$唯一确定，所以$2\delta$是一个定值)。在小波变换中，变换系数$W_f(a,b)$主要依赖于信号在$[b-a\Delta\Psi, b+a\Delta\Psi]$片段中的情况，时间宽度是$2a\Delta\Psi$，该时间宽度是随着尺度$a$变化而变化的，所以小波变换具有时间局部分析能力。

(6) 若用信号通过滤波器来解释，小波变换与短时傅里叶变换不同之处在于：对短时傅里叶变换来说，带通滤波器的带宽Δf与中心频率f无关；相反，小波变换带通滤波器的带宽Δf则正比于中心频率f，即

$$Q = \frac{\Delta f}{f} = C \qquad C\text{为常数}$$

亦即滤波器有一个恒定的相对带宽，称之为等Q结构(Q为滤波器的品质因数，且有$Q=\frac{\text{中心频率}}{\text{带宽}}$)。

1.2 常用小波函数介绍

与标准傅里叶变换相比，小波分析中所用到的小波函数具有不唯一性，即小波函数$\Psi(x)$具有多样性。但小波分析在工程应用中，一个十分重要的问题是最优小波基的选择问题，这是因为用不同的小波基分析同一个问题会产生不同的结果。目前主要是通过用小波分析方法处理信号的结果与理论结果的误差来判定小波基的好坏，并由此选定小波基。

根据不同的标准，小波函数具有不同的类型，这些标准通常有：

(1) Ψ、$\hat{\Psi}$、ϕ和$\hat{\phi}$的支撑长度。即当时间或频率趋向无穷大时，Ψ、$\hat{\Psi}$、ϕ和$\hat{\phi}$从一个有限值收敛到0的速度。

(2) 对称性。它在图像处理中对于避免移相是非常有用的。

(3) Ψ和ϕ(如果存在的情况)的消失矩阶数。它对于压缩是非常有用的。

(4) 正则性。它对信号或图像的重构获得较好的平滑效果是非常有用的。

但在众多小波基函数(也称核函数)的家族中，有一些小波函数被实践证明是非常有用的。我们可以通过waveinfo函数获得工具箱中的小波函数的主要性质，小波函数Ψ和尺度函数ϕ可以通过wavefun函数计算，滤波器可以通过wfilters函数产生。在本节中，我们主要介绍MATLAB中常用到的小波函数。

1.2.1 Haar小波

Haar函数是在小波分析中最早用到的一个具有紧支撑的正交小波函数，同时也是最简单的一个函数(如图1.3所示)。Haar函数的定义为

$$\Psi_H = \begin{cases} 1 & 0 \leqslant x \leqslant 1/2 \\ -1 & 1/2 \leqslant x < 1 \\ 0 & \text{其它} \end{cases} \tag{1.11}$$

在MATLAB中，可以输入命令waveinfo('haar')获得haar函数的一些主要性质。

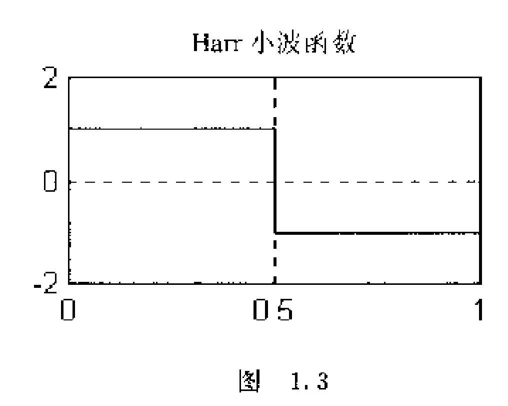


图　1.3

1.2.2　Daubechies(db*N*)小波系

Daubechies 函数是由世界著名的小波分析学者 Inrid Daubechies 构造的小波函数，除了 db1(即 haar 小波)外，其它小波没有明确的表达式，但转换函数 h 的平方模是很明确的。

· 假设 $P(y)=\sum_{k=0}^{N-1}C_k^{N-1+k}y^k$，其中，$C_k^{N-1+k}$ 为二项式的系数，则有

$$|m_0(\omega)|^2=(\cos^2\frac{\omega}{2})^N P(\sin^2\frac{\omega}{2}) \tag{1.12}$$

其中

$$m_0(\omega)=\frac{1}{\sqrt{2}}\sum_{k=0}^{2N-1}h_k e^{-ik\omega}$$

· 小波函数 Ψ 和尺度函数 ϕ 的有效支撑长度为 $2N-1$，小波函数 Ψ 的消失矩阶数为 N。

· db$N$ 大多数不具有对称性；对于有些小波函数，不对称性是非常明显的。

· 正则性随着序号 N 的增加而增加。

· 函数具有正交性。

在这里，我们画出 db4 和 db8 小波的尺度函数、小波函数、分解滤波器和重构滤波器的图形。如图 1.4 所示。

Daubechies 小波函数提供了比 Haar 函数更有效的分析和综合。Daubechies 系中的小波基记为 db$N$，N 为序号，且 $N=1，2，\cdots，10$。

在 MATLAB 中，可以输入命令 waveinfo('db')获得 Daubechies 函数的一些主要性质。

1.2.3　Biorthogonal (biorNr.Nd)小波系

Biorthogonal 函数系的主要特性体现在具有线性相位性，它主要应用在信号与图像的重构中。通常的用法是采用一个函数进行分解，用另外一个小波函数进行重构。Biorthogonal 函数系通常表示为 biorNr.Nd 的形式：

Nr＝1　　Nd＝1,3,5

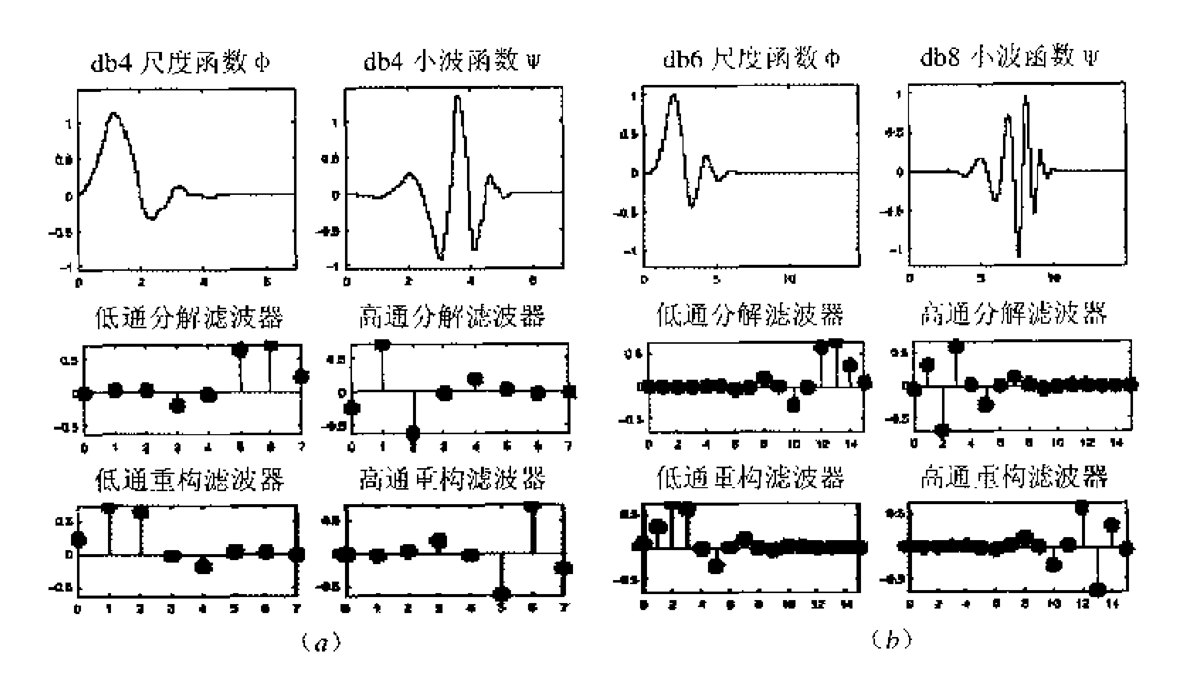
Nr＝2　　Nd＝2,4,6,8

Nr＝3　　Nd＝1,3,5,7,9

Nr＝4　　Nd＝4

Nr＝5　　Nd＝5

Nr＝6　　Nd＝8

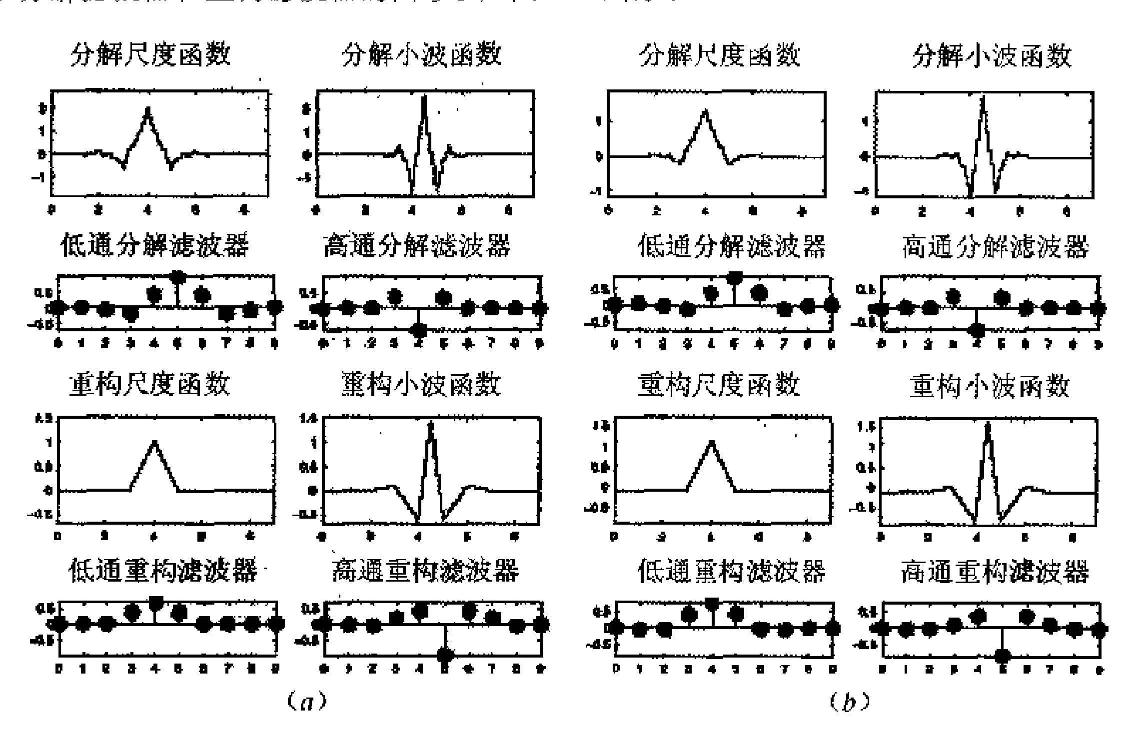


(a)　　(b)

图 1.4

其中，r 表示重构(Reconstruction)，d 表示分解(Decomposition)。

在这里，我们画出 bior2.4 和 bior4.4 小波(分别用于分解与重构)的尺度函数、小波函数、分解滤波器和重构滤波器的图形。如图 1.5 所示。



(a)　　(b)

图 1.5

在 MATLAB 中，可以输入命令 waveinfo('bior')获得该函数的主要性质。

1.2.4 Coiflet(coif$N$)小波系

coiflet 函数也是由 Daubechies 构造的一个小波函数，它具有 coif$N$($N=1,2,3,4,5$)这一系列。coiflet 具有比 db$N$ 更好的对称性。从支撑长度的角度看，coif$N$ 具有和 db3N 及 sym3N 相同的支撑长度；从消失矩的数目来看，coif$N$ 具有和 db2N 及 sym2N 相同的消失矩数目。

在这里，我们画出 coif3 和 coif5 小波的尺度函数、小波函数、分解滤波器和重构滤波器的图形。如图 1.6 所示。

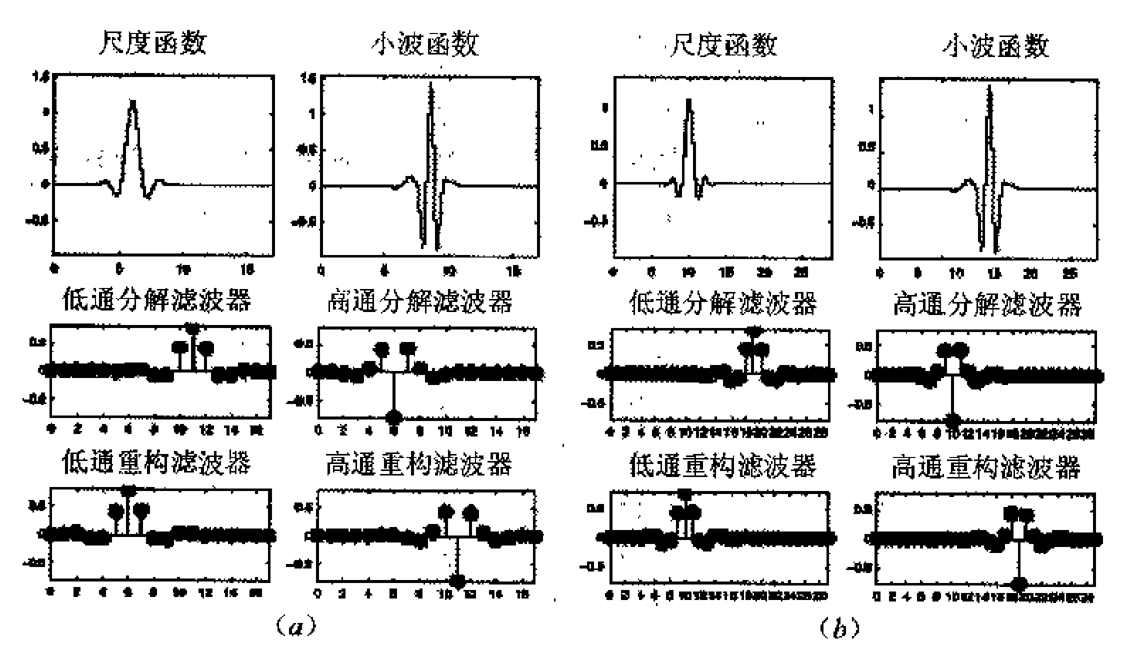


图 1.6

在 MATLAB 中，可以输入命令 waveinfo('coif')获得该函数的主要性质。

1.2.5 SymletsA(sym$N$)小波系

Symlets 函数系是由 Daubechies 提出的近似对称的小波函数，它是对 db 函数的一种改进。Symlets 函数系通常表示为 sym$N$($N=2,3,\cdots,8$)的形式。

在这里，我们画出 sym4 和 sym8 小波的尺度函数、小波函数、分解滤波器和重构滤波器的图形。如图 1.7 所示。

在 MATLAB 中，可以输入命令 waveinfo('sym') 获得该函数的主要性质。

1.2.6 Morlet(morl)小波

Morlet 函数定义为

$$\Psi(x) = Ce^{-x^2/2}\cos 5x \tag{1.13}$$

它的尺度函数不存在，且不具有正交性。

在 MATLAB 中，可以输入命令 waveinfo('morl')获得该函数的主要性质如图 1.8 所示。

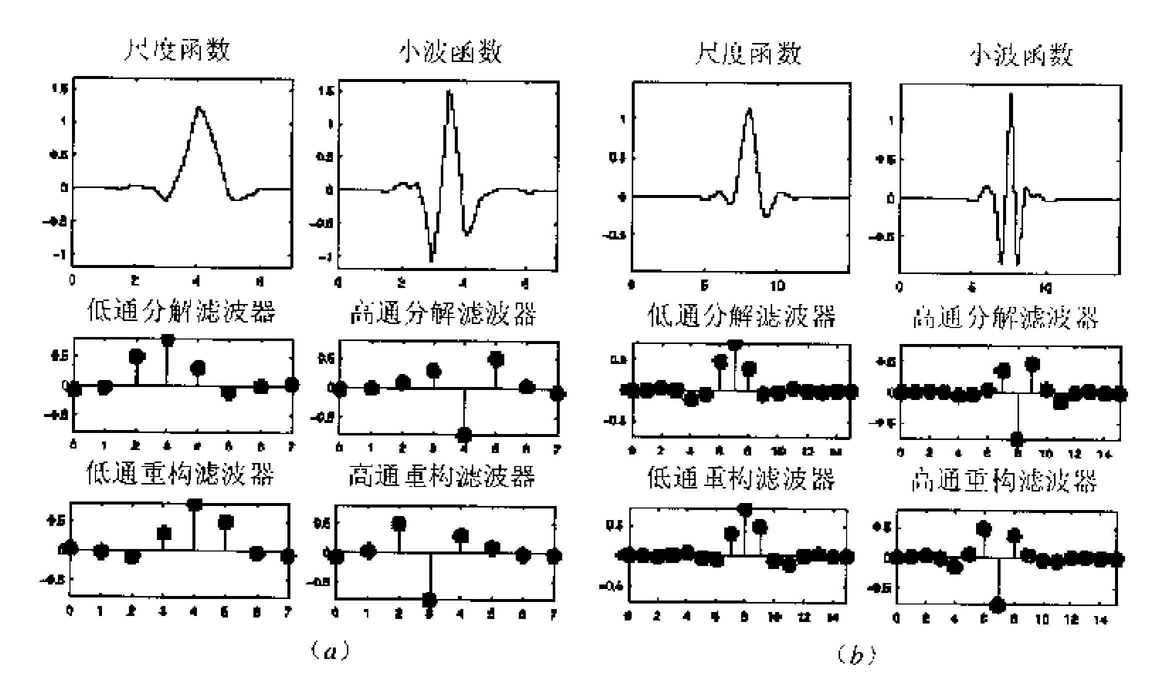


图 1.7

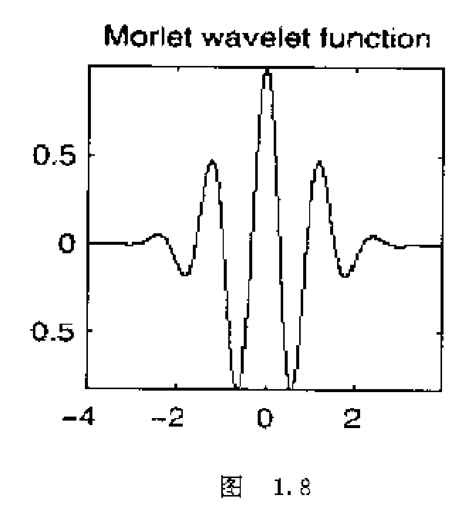


图 1.8

1.2.7 Mexican Hat(mexh)小波

Mexican Hat 函数为

$$\Psi(x) = \frac{2}{\sqrt{3}} \pi^{-1/4}(1 - x^2)\mathrm{e}^{-x^2/2} \tag{1.14}$$

它是 Gauss 函数的二阶导数，因为它像墨西哥帽的截面，所以有时称这个函数为墨西哥帽函数(如图 1.9 所示)。墨西哥帽函数在时间域与频率域都有很好的局部化，并且满足

$$\int_{-\infty}^{\infty} \Psi(x)\mathrm{d}x = 0 \tag{1.15}$$

由于它的尺度函数不存在，所以不具有正交性。

在 MATLAB 中，可以输入命令 waveinfo('mexh')获得该函数的主要性质。



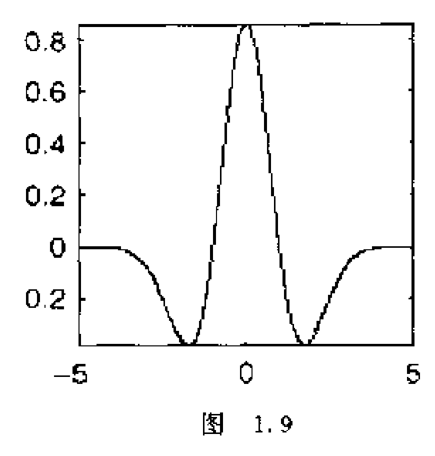


图　1.9

1.2.8 Meyer 函数

Meyer 小波的小波函数 Ψ 和尺度函数 φ 都是在频率域中进行定义的，是具有紧支撑的正交小波。

$$\hat{\Psi}(\omega)=\begin{cases}(2\pi)^{-1/2}\mathrm{e}^{\mathrm{i}\omega/2}\sin\left(\frac{\pi}{2}v\left(\frac{3}{2\pi}|\bar{\omega}|-1\right)\right) & \frac{2\pi}{3}\leqslant|\omega|\leqslant\frac{4\pi}{3}\\ (2\pi)^{-1/2}\mathrm{e}^{\mathrm{i}\omega/2}\cos\left(\frac{\pi}{2}v\left(\frac{3}{2\pi}|\bar{\omega}|-1\right)\right) & \frac{4\pi}{3}\leqslant|\omega|\leqslant\frac{8\pi}{3}\\ 0 & |\bar{\omega}|\notin\left[\frac{2\pi}{3},\frac{8\pi}{3}\right]\end{cases}\tag{1.16}$$

其中，$v(a)$为构造 Meyer 小波的辅助函数，且有

$$v(a)=a^4(35-84a+70a^2-20a^3)\qquad a\in[0,1]\tag{1.17}$$

$$\hat{\phi}(\omega)=\begin{cases}(2\pi)^{-1/2} & |\omega|\leqslant\frac{2\pi}{3}\\ (2\pi)^{-1/2}\cos\left(\frac{\pi}{2}v\left(\frac{3}{2\pi}|\bar{\omega}|-1\right)\right) & \frac{2\pi}{3}\leqslant\bar{\omega}\leqslant\frac{4\pi}{3}\\ 0 & |\omega|>\frac{4\pi}{3}\end{cases}\tag{1.18}$$

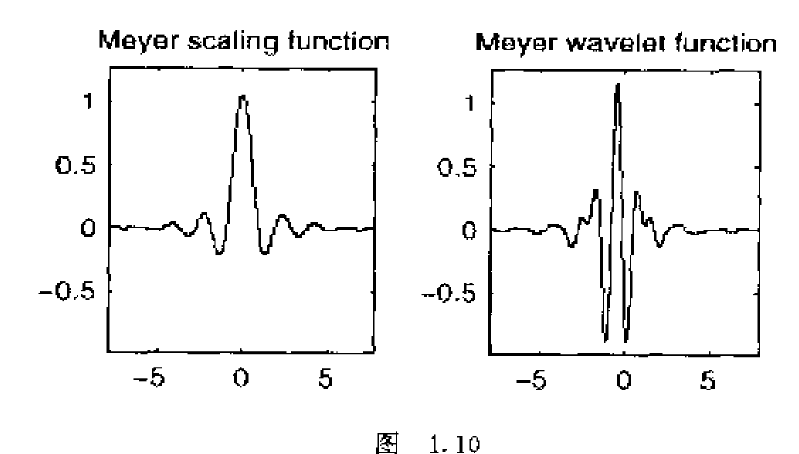


图　1.10

在 MATLAB 中，可以输入命令 waveinfo('meyer')获得该函数的主要属性。

1.2.9 Battle－Lemarie 小波

该小波在 MATLAB 工具箱中不存在，但它也是我们常用到的一个小波函数，它具有两种形式，一种具有确定的正交性，一种不具有确定的正交性。当 $N=1$ 时，尺度函数是线性样条函数；当 $N=2$ 时，尺度函数是具有有限支撑的 B 样条函数。更一般的情况，对于一个 N 次 B 样条小波，尺度函数为

$$\phi(\bar{\omega}) = (2\pi)^{-1/2} e^{-ik\omega/2}\left[\frac{\sin(\bar{\omega}/2)}{\bar{\omega}/2}\right]^{N-1} \tag{1.19}$$

当 N 为奇数时，$k=0$；当 N 为偶数时，$k=1$。式(1.18)可以用来构造滤波器。

它的双尺度关系为

$$\phi(x) = \begin{cases} 2^{-2M}\sum_{j=0}^{2M+1} C_j^{2M+1}\,\phi(2x-M-1+j) & N=2M \\ 2^{-2M-1}\sum_{j=0}^{2M-2} C_j^{2M+2}\,\phi(2x-M-1+j) & N=2M+1 \end{cases} \tag{1.20}$$

当 N 为偶数时，ϕ 是对称的，$x=1/2$；当 N 为奇数时，ϕ 是反对称的，$x=0$。

下面，我们将 MATLAB 工具箱中八个小波(或小波系)的一些主要性质加以对比，见表 1－1。

表 1－1　MATLAB 工具箱中八个小波(或小波系)的主要性质

小波函数	Haar	Daubechies	Biorthgonal	Coiflets	Symlets	Morlet	Mexican hat	Meyer
小波缩写名	haar	db	bior	coif	sym	morl	mexh	meyr
表示形式	haar	db$N$	biorNr. Nd	coif$N$	sym$N$	morl	mexh	meyr
举例	haar	db3	bior2. 4	coif3	sym2，sym4	morl	mexh	meyr
正交性	有	有	无	有	有	无	无	有
双正交性	有	有	有	有	有	无	无	有
紧支撑性	有	有	有	有	有	无	无	无
连续小波变换	可以	可以	可以	可以	可以	可以	可以	可以
离散小波变换	可以	可以	可以	可以	可以	不可以	不可以	可以，但无 FWT
支撑长度	1	$2N-1$	重构：2Nr＋1 分解：2Nd＋1	$6N-1$	$2N-1$	有限长度	有限长度	有限长度
滤波器长度	2	$2N$	max(2Nr，2Nd)＋2	$6N$	$2N$	[－4，4]	[－5，5]	[－8，8]
对称性	对称	近似对称	不对称	近似对称	近似对称	对称	对称	对称
小波函数 Ψ 消失矩阶数	1	N	Nr－1	$2N$	N	－	－	－
尺度函数 ϕ 消失矩阶数	－	－		$2N-1$	－	－	－	－

1.3　连续小波变换

1.3　一维连续小波变换

定义 1.3　设 $\Psi(t)\in L^2(R)$，其傅里叶变换为 $\hat{\Psi}(\bar{\omega})$，当 $\hat{\Psi}(\omega)$满足允许条件(完全重构条件或恒等分辨条件)

$$C_\Psi = \int_R \frac{|\hat{\Psi}(\omega)|^2}{|\omega|} \mathrm{d}\omega < \infty \tag{1.21}$$

时，我们称 $\Psi(t)$为一个基本小波或母小波(Mother Wavelet)。将母函数 $\Psi(t)$经伸缩和平移后得

$$\Psi_{a,b}(t) = \frac{1}{\sqrt{|a|}}\Psi\left(\frac{t-b}{a}\right) \qquad a,b \in R;\ a \neq 0 \tag{1.22}$$

称其为一个小波序列。其中 a 为伸缩因子，b 为平移因子。

对于任意的函数 $f(t)\in L^2(R)$的连续小波变换为

$$W_f(a,b) = \langle f,\Psi_{a,b}\rangle = |a|^{-1/2}\int_R f(t)\,\overline{\Psi\left(\frac{t-b}{a}\right)}\mathrm{d}t \tag{1.23}$$

其重构公式(逆变换)为

$$f(t) = \frac{1}{C_\Psi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\frac{1}{a^2}W_f(a,b)\Psi\left(\frac{t-b}{a}\right)\mathrm{d}a\mathrm{d}b \tag{1.24}$$

由于基小波 $\Psi(t)$生成的小波 $\Psi_{a,b}(t)$在小波变换中对被分析的信号起着观测窗的作用，所以 $\Psi(t)$还应该满足一般函数的约束条件

$$\int_{-\infty}^{\infty}|\Psi(t)|\mathrm{d}t < \infty \tag{1.25}$$

故 $\hat{\Psi}(\omega)$是一个连续函数。这意味着，为了满足完全重构条件式(1.21)，$\hat{\Psi}(\omega)$在原点必须等于 0，即

$$\hat{\Psi}(0) = \int_{-\infty}^{\infty}\Psi(t)\mathrm{d}t = 0$$

为了使信号重构的实现在数值上是稳定的，除了完全重构条件外，还要求小波 $\Psi(t)$的傅里叶变换满足下面的稳定性条件：

$$A \leqslant \sum_{-\infty}^{\infty}|\hat{\Psi}(2^{-j}\omega)|^2 \leqslant B \tag{1.26}$$

式中，$0<A\leqslant B<\infty$。

从稳定性条件可以引出一个重要的概念。

定义 1.4(对偶小波)　若小波 $\Psi(t)$满足稳定性条件(1.26)式，则定义一个对偶小波 $\tilde{\Psi}(t)$，其傅里叶变换 $\hat{\tilde{\Psi}}(\omega)$由下式给出：

$$\hat{\tilde{\Psi}}(\omega) = \frac{\hat{\Psi}^*(\omega)}{\sum_{j=-\infty}^{\infty}|\hat{\Psi}(2^{-j}\omega)|^2} \tag{1.27}$$

注意，稳定性条件(1.26)式实际上是对(1.27)式分母的约束条件，它的作用是保证对偶小波的傅里叶变换存在稳定性。

值得指出的是，一个小波的对偶小波一般不是唯一的，然而，在实际应用中，我们又总是希望它们是唯一对应的。因此，寻找具有唯一对偶小波的合适小波也就成为小波分析中最基本的问题。

连续小波变换具有以下重要性质：

① 线性性：一个多分量信号的小波变换等于各个分量的小波变换之和。

② 平移不变性：若 $f(t)$ 的小波变换为 $W_f(a,b)$，则 $f(t-\tau)$ 的小波变换为

$$W_f(a,b-\tau)$$

③ 伸缩共变性：若 $f(t)$ 的小波变换为 $W_f(a,b)$，则 $f(ct)$ 的小波变换为

$$\frac{1}{\sqrt{c}}W_f(ca,cb) \qquad c>0$$

④ 自相似性：对应不同尺度参数 a 和不同平移参数 b 的连续小波变换之间是自相似的。

⑤ 冗余性：连续小波变换中存在信息表述的冗余度(redundancy)。

小波变换的冗余性事实上也是自相似性的直接反映，它主要表现在以下两个方面：

① 由连续小波变换恢复原信号的重构分式不是唯一的。也就是说，信号 $f(t)$ 的小波变换与小波重构不存在一一对应关系，而傅里叶变换与傅里叶反变换是一一对应的。

② 小波变换的核函数即小波函数 $\Psi_{a,b}(t)$ 存在许多可能的选择(例如，它们可以是非正交小波、正交小波、双正交小波，甚至允许是彼此线性相关的)。

小波变换在不同的 (a,b) 之间的相关性增加了分析和解释小波变换结果的困难，因此，小波变换的冗余度应尽可能减小，它是小波分析中的主要问题之一。在MATLAB中，可以用cwt函数实现对信号的连续小波变换。

例2 已知一信号 $f(t)=3\sin 100\pi t+2\sin 68\pi t+5\cos 72\pi t$ 的信号，且该信号混有白噪声，对该信号进行连续小波变换。小波函数取db3，尺度为1、1.2、1.4、1.6、…3。其MATLAB程序如下：

```
t=0:0.01:1;
f=3*sin(100*pi*t)+2*sin(68*pi*t)+5*cos(72*pi*t)+randn(1,length(t));
coefs=cwt(f,[1:0.2:3],'db3','plot');
title('对不同的尺度小波变换系数值');
Ylabel('尺度');
Xlabel('时间');
```

输出结果(如图1.11所示)。

小波变换的系数用图1.11所示的灰度值图表征，横坐标表示变换系数的系号，纵坐标表示尺度，灰度颜色越深，表示系数的值越大。

1.3.2 高维连续小波变换

对 $f(t)\in L^2(R^n)(n>1)$，公式



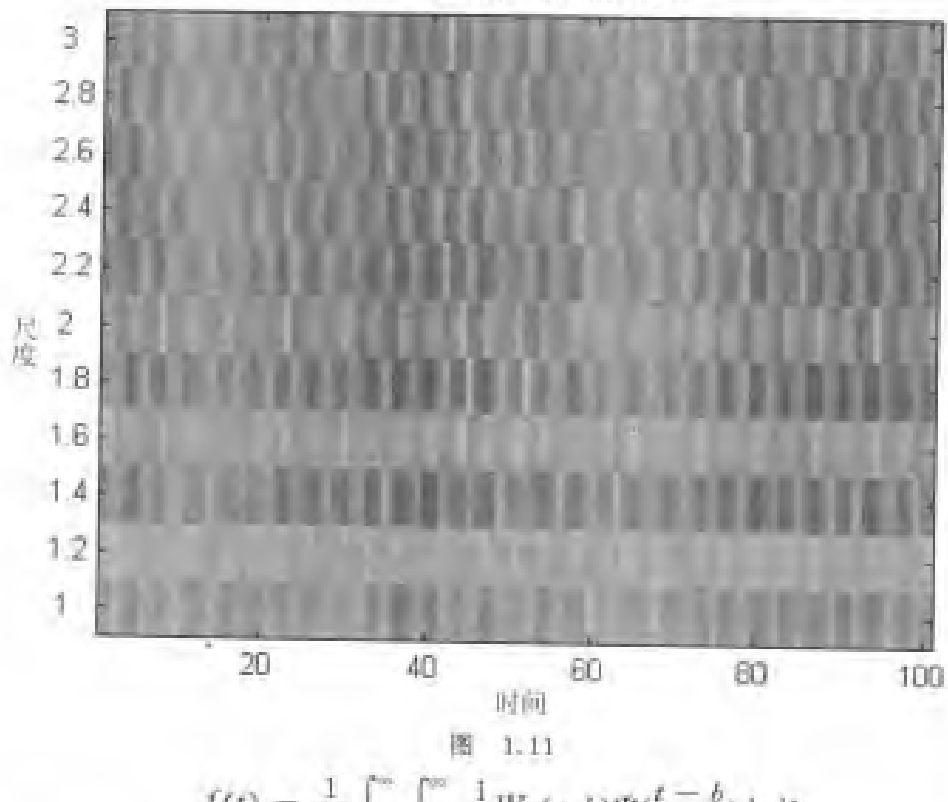


图 1.11

$$f(t) = \frac{1}{C_\Psi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\frac{1}{a^2}W_f(a,b)\Psi(\frac{t-b}{a})\mathrm{d}a\mathrm{d}b$$

存在几种扩展的可能性，一种可能性是选择小波 $f(t)\in L^2(R^n)$ 使其为球对称，其傅里叶变换也同样球对称，

$$\hat{\Psi}(\overline{\omega}) = \eta(|\overline{\omega}|) \tag{1.28}$$

并且其相容性条件变为

$$C_\Psi = (2\pi)^n\int_0^\infty |\eta(t)|^2\frac{\mathrm{d}t}{t} < \infty \tag{1.29}$$

对所有的 $f,g\in L^2(g^n)$。

$$\int_0^\infty \frac{\mathrm{d}a}{a^{n+1}}W_f(a,b)\overline{W}_g(a,b)\mathrm{d}b = C_\Psi < f, g > \tag{1.30}$$

这里，$W_f(a,b)=\langle f,\Psi^{a,b}\rangle$，$\Psi^{a,b}(t)=a^{-n/2}\Psi(\frac{t-b}{a})$，其中 $a\in R^+$，$a\neq 0$ 且 $b\in R^n$，公式(1.24)也可以写为

$$f = C_\Psi^{-1}\int_0^\infty\frac{\mathrm{d}a}{a^{n+1}}\int_{R^n}W_f(a,b)\Psi^{a,b}\,\mathrm{d}b \tag{1.31}$$

如果选择的小波 Ψ 不是球对称的，但可以用旋转进行同样的扩展与平移。例如，在二维时，可定义

$$\Psi^{a,b,\theta}(t) = a^{-1}\Psi(R_\theta^{-1}(\frac{t-b}{a})) \tag{1.32}$$

这里，$a>0$，$b\in R^2$，$R_\theta=\begin{pmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{pmatrix}$，相容条件变为

$$C_\Psi = 2(\pi)^2\int_0^\infty\frac{\mathrm{d}r}{r}\int_0^{2\pi}|\hat{\Psi}(r\cos\theta, r\sin\theta)|^2\mathrm{d}\theta < \infty \tag{1.33}$$

该等式对应的重构公式为

$$f = C_{\Psi}^{-1}\int_{0}^{\infty}\frac{\mathrm{d}a}{a^{3}}\int_{R^{2}}\mathrm{d}b\int_{0}^{2\pi}W_{f}(a,b,\theta)\Psi^{a,b,\theta}\mathrm{d}\theta \tag{1.34}$$

对于高于二维的情况，可以给出类似的结论。

在 MATLAB 中，系统只给出了关于一维和二维的小波变换函数，有关这些函数用法的详细说明见 2.2、2.3 两节。

1.4　离散小波变换

1.4.1　离散小波变换

在实际运用中，尤其是在计算机上实现时，连续小波必须加以离散化。因此，有必要讨论连续小波 $\Psi_{a,b}(t)$和连续小波变换 $W_f(a,b)$的离散化。需要强调指出的是，这一离散化都是针对连续的尺度参数 a 和连续平移参数 b 的，而不是针对时间变量 t 的。这一点与我们以前习惯的时间离散化不同，希望引起注意。

在连续小波中，考虑函数：

$$\Psi_{a,b}(t) = |a|^{-1/2}\Psi(\frac{t-b}{a}) \tag{1.35}$$

这里，$b\in R$，$a\in R_{+}$，且 $a\neq 0$，Ψ 是容许的，为方便起见，在离散化中，总限制 a 只取正值，这样相容性条件就变为

$$C_{\Psi} = \int_{0}^{\infty}\frac{|\hat{\Psi}(\bar{\omega})|}{|\bar{\omega}|}\mathrm{d}\bar{\omega} < \infty \tag{1.36}$$

通常，把连续小波变换中尺度参数 a 和平移参数 b 的离散化公式分别取作 $a=a_0^j$，$b=ka_0^jb_0$，这里 $j\in Z$，扩展步长 $a_0\neq 1$ 是固定值，为方便起见，总是假定 $a_0>1$(由于 m 可取正也可取负，所以这个假定无关紧要)。所以对应的离散小波函数 $\Psi_{j,k}(t)$即可写作

$$\Psi_{j,k}(t) = a_0^{-j/2}\Psi(\frac{t-ka_0^jb_0}{a_0^j}) = a_0^{-j/2}\Psi(a_0^{-j}t-kb_0) \tag{1.37}$$

而离散化小波变换系数则可表示为

$$C_{j,k} = \int_{-\infty}^{\infty}f(t)\Psi_{j,k}^{*}(t)\mathrm{d}t = \langle f,\Psi_{j,k}\rangle \tag{1.38}$$

其重构公式为

$$f(t) = C\sum_{-\infty}^{\infty}\sum_{-\infty}^{\infty}C_{j,k}\Psi_{j,k}(t) \tag{1.39}$$

C 是一个与信号无关的常数。

然而，怎样选择 a_0 和 b_0，才能够保证重构信号的精度呢？显然，网格点应尽可能密(即 a_0 和 b_0 尽可能小)，因为如果网格点越稀疏，使用的小波函数 $\Psi_{j,k}(t)$和离散小波系数 $C_{j,k}$ 就越少，信号重构的精确度也就会越低。

在 MATLAB 中，离散小波变换可以用 dwt，dwt2 函数实现。

1.4.2　二进制小波变换

上面是对尺度参数 a 和平移参数 b 进行离散化的要求。为了使小波变换具有可变化的

时间和频率分辨率，适应待分析信号的非平稳性，我们很自然地需要改变 a 和 b 的大小，以使小波变换具有"变焦距"的功能。换言之，在实际中采用的是动态的采样网格。最常用的是二进制的动态采样网格，即 $a_0=2$，$b_0=1$，每个网格点对应的尺度为 2^j，而平移为 2^jk。由此得到的小波

$$\Psi_{j,k}(t) = 2^{-j/2}\Psi(2^{-j}t - k) \qquad j,k \in Z \tag{1.40}$$

称为二进小波(Dyadic Wavelet)。

二进小波对信号的分析具有变焦距的作用。假定有一放大倍数 2^{-j}，它对应为观测到信号的某部分内容。如果想进一步观看信号更小的细节，就需要增加放大倍数即减小 j 值；反之，若想了解信号更粗的内容，则可以减小放大倍数，即加大 j 值。在这个意义上，小波变换被称为数学显微镜。

定义 1.5　设函数 $\Psi_{j,k}(t)\in L^2(R)$，如果存在两个常数 A、B，且 $0<A<B<\infty$ 使得稳定性条件几乎处处成立，即

$$A \leqslant \sum_{j\in Z} |\hat{\Psi}(2^{-j}\omega)|^2 \leqslant B \tag{1.41}$$

则 $\Psi_{j,k}(x)$ 为一个二进小波。若 $A=B$，则称为最稳定条件。而函数序列 $\{W_{2^j}f(k)\}_{k\in Z}$ 叫做 f 的二进小波变换，其中

$$W_{2^j}f(k) = \langle f(t), \Psi_{2^j}(k)\rangle = \frac{1}{2^j}\int_R f(t)\Psi^*(2^{-j}t - k)\mathrm{d}t$$

上式相应的逆变换为

$$f(t) = \sum_{j\in Z} W_{2^j}f(k) * \Psi_{2^j}(t) = \sum_{j\in Z}\int W_{2^j}f(x)\Psi_{2^j}(2^{-j}t - k)\mathrm{d}k \tag{1.42}$$

二进小波不同于连续小波的离散小波，它只是对尺度参数进行了离散化，而对时间域上的平移参量保持连续变化，因此二进小波不破坏信号在时间域上的平移不变量，这也正是它同正交小波基相比所具有的独特优点。

1.5　多分辨分析

Meyer 于 1986 年创造性地构造出具有一定衰减性的光滑函数，其二进制伸缩与平移构成 $L^2(R)$ 的规范正交基，才使小波得到真正的发展。1988 年 S. Mallat 在构造正交小波基时提出了多分辨分析(Multi - Resolution Analysis)的概念，从空间的概念上形象地说明了小波的多分辨率特性，将此之前的所有正交小波基的构造法统一起来，给出了正交小波的构造方法以及正交小波变换的快速算法，即 Mallat 算法。Mallat 算法在小波分析中的地位相当于快速傅里叶变换算法在经典傅里叶分析中的地位。

关于多分辨分析的理解，我们在这里以一个三层的分解进行说明，其小波分解树如图 1.12 所示。

从图 1.12 可以明显看出，多分辨分析只是对低频部分进行进一步分解，而高频部分则不予以考虑。分解具有关系：S＝A3＋D3＋D2＋D1。另外强调一点，这里只是以一个层分解进行说明，如果要进行进一步的分解，则可以把低频部分 A3 分解成低频部分 $A4$ 和高频部分 D4，以下再分解依此类推。

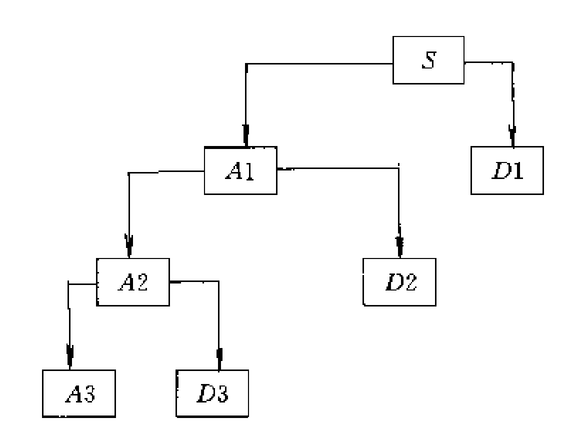


图 1.12　三层多分辨分析树结构图

在理解多分辨分析以及下一节的小波包分析，我们必须牢牢把握一点，即分解的最终目的是力求构造一个在频率上高度逼近 $L^2(R)$空间的正交小波基(或正交小波包基)，这些频率分辨率不同的正交小波基相当于带宽各异的带通滤波器。从上面的多分辨分析树型结构图 1.12 可以看出，多分辨分析只对低频空间进行进一步的分解，使频率的分辨率变得越来越高。下面我们分析多分辨分析是如何构造正交小波基的。

定义 1.6　空间 $L^2(R)$中的多分辨分析是指 $L^2(R)$中满足如下条件的一个空间序列 $\{V_j\}_{j\in Z}$：

① 单调性：$V_j \subset V_{j+1}$，对任意 $j\in Z$。

② 逼近性：$\bigcap_{j\in Z} V_j=\{0\}$，$\mathrm{close}\{\bigcup_{-\infty}^{\infty} V_j\}=L^2(R)$。

③ 伸缩性：$f(t)\in V_j \Leftrightarrow f(2t)\in V_{j+1}$；伸缩性体现了尺度的变化、逼近正交小波函数的变化和空间的变化具有一致性。

④ 平移不变性：对任意 $k\in Z$，有 $\phi_j(2^{-j/2}t)\in V_j \Rightarrow \phi_j(2^{-j/2}t-k)\in V_j$。

⑤ Riesz 基存在性：存在 $\phi(t)\in V_0$，使得$\{\phi(2^{-j/2}t-k)\,|\,k\in Z\}$构成 V_j 的 Riesz 基。

对于条件⑤，可以证明，存在函数 $\phi(t)\in V_0$，使它的整数平移系$\{\phi(2^{-j/2}t-k)\,|\,k\in Z\}$构成 V_j 的规范正交基，我们称 $\phi(t)$为尺度函数(Scaling Function)。定义函数

$$\phi_{j,k}(t) = 2^{-j/2}\phi(2^{-j}t-k) \qquad j,k\in Z \tag{1.43}$$

则函数系$\{\phi_{j,k}(t)\,|\,k\in Z\}$是规范正交的。

设以 V_j 表示图 1.12 分解中的低频部分 A_j，W_j 表示分解中的高频部分 D_j，则 W_j 是 V_j 在 V_{j+1}中的正交补，即

$$V_j \oplus W_j = V_{j+1} \qquad j\in Z \tag{1.44}$$

显然

$$V_j \oplus W_j \oplus W_{j-1} \oplus \cdots \oplus W_{j+m} = V_{j+m} \tag{1.45}$$

则多分辨分析的子空间 V_0 可以用有限个子空间来逼近，即有

$$V_0 = V_1 \oplus W_1 = V_2 \oplus W_2 \oplus W_1 = \cdots = V_N \oplus W_N \oplus W_{N-1} \oplus \cdots \oplus W_2 \oplus W_1 \tag{1.46}$$

空间列$\{W_j\,|\,j\in Z\}$具有性质：

① $f(t)\in W_j \Rightarrow f(t-2^j n)\in W_j \qquad j,n\in Z$

② $f(t)\in W_j \Leftrightarrow f(2t)\in W_{j-1} \qquad j\in Z$

③ $P_{W_j f}\to 0$，当$|j|\to\infty$，对任意$f\in L^2(R)$和V_j一样，我们希望找出一个确定的函数$\Psi(t)\in W_0$，使得对每个$j\in Z$，函数系$\{\Psi_{j,n}|n\in Z\}$构成空间W_j的规范正交基，其中$\Psi_{j,n}(t)=2^{-j/2}\Psi(2^{-j}t-n)$。

若令$f_j\in V_j$代表分辨率为2^{-j}的函数$f\in L^2(R)$的逼近(即函数f的低频部分或“粗糙像”)，而$d_j\in W_j$代表逼近的误差(即函数f的高频部分或“细节”部分)，则式(1.46)意味着：

$$f_0=f_1+f_d=f_2+d_2+d_1=\cdots=f_N+d_N+d_{N-1}+\cdots+d_2+d_1 \tag{1.47}$$

注意到$f=f_0$，所以上式可简写为

$$f=f_N+\sum_{i=1}^{N}d_i \tag{1.48}$$

这表明，任何函数$f\in L^2(R)$都可以根据分辨率为2^{-N}时f的低频部分(“粗糙像”)和分辨率$2^{-j}(1\leqslant j\leqslant N)$下$f$的高频部分(“细节”部分)完全重构，这恰好是著名 Mallat 塔式重构算法的思想。

从包容关系$V_0\subset V_{-1}$，我们很容易得到尺度函数$\phi(t)$的一个极为有用的性质。注意到$\phi_{0,0}(t)\in V_0\in V_{-1}$，所以$\phi(t)=\phi_{0,0}(t)$可以用$V_{-1}$子空间的基函数$\phi_{-1,k}(t)=2^{1/2}\phi(2t-k)$展开，令展开系数为$h_k$，则

$$\phi(t)=\sqrt{2}\sum_{-\infty}^{\infty}h(k)\phi(2t-k) \tag{1.49}$$

这就是尺度函数的双尺度方程。

另一方面，由于$V_{-1}=V_0\oplus W_0$，故$\Psi(t)=\Psi_{0,0}(t)\in W_0\in W_{-1}$，这意味着小波基函数$\Psi(t)$可以用$V_{-1}$子空间的正交基$\phi_{-1,k}(t)=2^{1/2}\phi(2t-k)$展开，令展开系数为$g_k$，即有

$$\Psi(t)=\sqrt{2}\sum_{-\infty}^{\infty}g(k)\phi(2t-k) \tag{1.50}$$

这就是小波函数的双尺度方程。

双尺度方程式(1.49)和式(1.50)表明，小波基$\Psi_{j,k}(t)$可由尺度函数$\phi(t)$的平移和伸缩的线性组合获得，其构造归结为滤波器$H(w)$($h(k)$的频域表示)和$G(w)$($g(k)$的频域表示)的设计。

由$\Psi(t)$的二进制伸缩平移形成的空间为

$$W_j=\overline{\mathrm{span}_{L^2}\{\Psi_{j,n}(t)=2^{j/2}\Psi(2^jt-n),\ n\in Z\}}$$

则

① $W_j\perp V_j$，$W_j\oplus V_j=V_{j+1}$，从而$L^2(R)=\overline{\bigoplus_{j\in Z}W_j}$，$W_j\perp W_{j'}$，$j\neq j'$。

② $\{\Psi_{j,n}\}_{n\in Z}$是W_j中的标准正交基，从而$\{\Psi_{j,n}\}_{j,n\in Z}$是$L^2(R)$中的标准正交基，即小波正交基。

综合以上分析，我们可以归纳出为了使$\phi_{j,k}(t)=2^{-j/2}\phi(2^{-j}t-k)$构成$V_j$子空间的正交基，生成元$\phi(t)$(尺度函数)应该具有下列基本性质：

① 尺度函数的容许条件，$\int_{-\infty}^{\infty}\phi(t)\mathrm{d}t=1$。

② 能量归一化条件：$\|\phi\|_2^2=1$。

③ 尺度函数$\phi(t)$具有正交性，即$\langle\phi(t-l),\phi(t-k)\rangle=\delta(k-l)$，$\forall\ k,l\in Z$。

④ 尺度函数 $\phi(t)$与基小波函数 $\Psi(t)$正交，即$\langle\phi(t), \Psi(t)\rangle=0$。

⑤ 跨尺度的尺度函数 $\phi(t)$与 $\phi(2t)$相关，满足双尺度方程(1.49)。

⑥ 基小波函数 $\Psi(t)$和 $\phi(2t)$相关，即满足小波函数的双尺度方程(1.50)。

将尺度函数的容许条件与小波的容许条件 $\int_{-\infty}^{\infty}\Psi(t)\mathrm{d}t=0$ 作一比较知，尺度函数的傅里叶变换 $\hat{\phi}(w)$具有低通滤波特性(相当于一个低通滤波器)，而小波的傅里叶变换 $\hat{\Psi}(w)$则具有高通滤波特性(相当于一个带通滤波器)。

除了以上要求外，尺度函数 $\phi(t)$还应该是 R 域上的实值函数，并且是 r 次可微分的，其导数连续，具有足够的下降速度，即 $\phi(t)$应满足：

$$|\phi^{(k)}(t)|\leqslant C_{pk}(1+|t|)^{p}\qquad k=0,1,\cdots,r;\qquad p\in Z;t\in R$$

1.6 小波包分析

短时傅里叶变换对信号的频带划分是线性等间隔的。多分辨分析可以对信号进行有效的时频分解，但由于其尺度是按二进制变化的，所以在高频频段其频率分辨率较差，而在低频频段其时间分辨率较差，即对信号的频带进行指数等间隔划分(具有等 Q 结构)。小波包分析(Wavelet Packet Analysis)能够为信号提供一种更加精细的分析方法，它将频带进行多层次划分，对多分辨分析没有细分的高频部分进一步分解，并能够根据被分析信号的特征，自适应地选择相应频带，使之与信号频谱相匹配，从而提高了时—频分辨率，因此小波包具有更广泛的应用价值。

关于小波包分析的理解，我们在这里以一个三层的分解进行说明，其小波包分解树如图 1.13 所示。

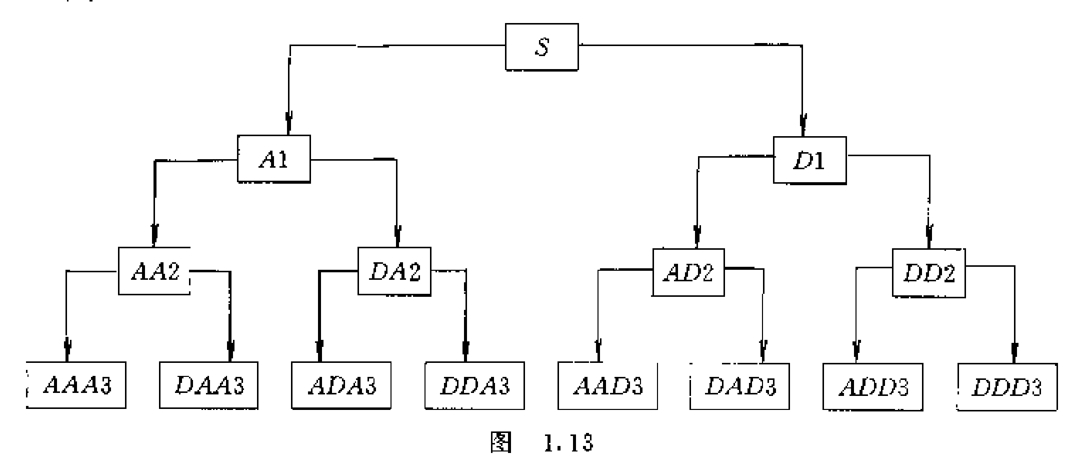


图 1.13

上图中，A 表示低频，D 表示高频，末尾的序号数表示小波包分解的层数(也即尺度数)。在 MALTAB 中关于小波包分析的函数详见 2.5 节。分解具有关系：$S=AAA3+DAA3+ADA3+DDA3+AAD3+DAD3+ADD3+DDD3$。

1.6.1 小波包的定义

在多分辨分析中，$L^2(R)=\bigoplus_{j\in Z}W_j$，表明多分辨分析是按照不同的尺度因子 j 把 Hilbert 空间 $L^2(R)$分解为所有子空间 $W_j(j\in Z)$的正交和的。其中，W_j 为小波函数 $\Psi(t)$的闭包

（小波子空间）。现在，我们希望进一步对小波子空间 W_j 按照二进制分式进行频率的细分，以达到提高频率分辨率的目的。

一种自然的做法是将尺度子空间 V_j 和小波子空间 W_j 用一个新的子空间 U_j^n 统一起来表征，若令

$$\left.\begin{aligned} U_j^0 &= V_j \\ U_j^1 &= W_j \end{aligned}\right\} \quad j \in Z \tag{1.51}$$

则 Hilbert 空间的正交分解 $V_{j+1}=V_j \oplus W_j$ 即可用 U_j^n 的分解统一为

$$U_{j+1}^0 = U_j^0 \oplus U_j^1 \qquad j \in Z \tag{1.52}$$

定义子空间 U_j^n 是函数 $u_n(t)$ 的闭包空间，而 U_j^{2n} 是函数 $u_{2n}(t)$ 的闭包空间，并令 $u_n(t)$ 满足下面的双尺度方程：

$$\left.\begin{aligned} u_{2n}(t) &= \sqrt{2} \sum_{k \in Z} h(k) u_n(2t-k) \\ u_{2n-1}(t) &= \sqrt{2} \sum_{k \in Z} g(k) u_n(2t-k) \end{aligned}\right\} \tag{1.53}$$

式中，$g(k)=(-1)^k h(1-k)$，即两系数也具有正交关系。当 $n=0$ 时，以上两式直接给出

$$\left.\begin{aligned} u_0(t) &= \sum_{k \in Z} h_k u_0(2t \quad k) \\ u_1(t) &= \sum_{k \in Z} g_k u_0(2t-k) \end{aligned}\right\} \tag{1.54}$$

与在多分辨分析中，$\phi(t)$ 和 $\Psi(t)$ 满足双尺度方程：

$$\left.\begin{aligned} \phi(t) &= \sum_{k \in Z} h_k \phi(2t-k) \qquad \{h_k\}_{k \in Z} \in l^2 \\ \Psi(t) &= \sum_{k \in Z} g_k \phi(2t-k) \qquad \{g_k\}_{k \in Z} \in l^2 \end{aligned}\right\} \tag{1.55}$$

相比较，$u_0(t)$ 和 $u_1(t)$ 分别退化为尺度函数 $\phi(t)$ 和小波基函数 $\Psi(t)$。式(1.54)是式(1.52)的等价表示。把这种等价表示推广到 $n \in Z_+$（非负整数）的情况，即得(1.53)的等价表示为

$$U_{j-1}^n = U_j^n \oplus U_j^{2n+1} \qquad j \in Z;\ n \in Z_+ \tag{1.56}$$

定义 1.7（小波包）　由式(1.53)构造的序列 $\{u_n(t)\}$（其中 $n \in Z_+$）称为由基函数 $u_0(t) = \phi(t)$ 确定的正交小波包。当 $n=0$ 时，即为(1.54)式的情况。

由于 $\phi(t)$ 由 h_k 唯一确定，所以又称 $\{u_n(t)\}_{n \in Z}$ 为关于序列 $\{h_k\}$ 的正交小波包。

1.6.2　小波包的性质

定理 1.1　设非负整数 n 的二进制表示为

$$n = \sum_{i=1}^{\infty} \varepsilon_i 2^{i-1} \qquad \varepsilon_i = 0 \text{ 或 } 1$$

则小波包 $\hat{u}_n(\omega)$ 的傅里叶变换由下式给出：

$$\hat{u}_n(\bar{\omega}) = \prod_{i=1}^{+\infty} m_{\varepsilon_i}(\omega/2^j)$$

式中

$$m_0(\bar{\omega}) = H(\omega) = \frac{1}{\sqrt{2}} \sum_{k=-\infty}^{\infty} h(k) e^{-jk\omega}$$

$$m_1(\overline{\omega}) = G(\omega) = \frac{1}{\sqrt{2}} \sum_{k=-\infty}^{\infty} g(k) e^{-jk\omega}$$

定理 1.2 设$\{u_n(t)\}_{n\in Z}$是正交尺度函数$\phi(t)$的正交小波包，则$\langle u_n(t-k), u_n(t-l)\rangle = \delta_{kl}$，即$\{u_n(t)\}_{n\in Z}$构成$L^2(R)$的规范正交基。

1.6.3 小波包的空间分解

令$\{u_n(t)\}_{n\in Z}$是关于h_k的小波包族，考虑用下列方式生成子空间族。现在令$n=1, 2, \cdots$；$j=1, 2, \cdots$，并对(1.52)式作迭代分解，则有

$$W_j = U_j^1 = U_{j-1}^2 \oplus U_{j-1}^3$$
$$U_{j-1}^2 = U_{j-2}^4 \oplus U_{j-2}^5,\ U_{j-2}^5 = U_{j-2}^6 \oplus U_{j-2}^7$$
$$\cdots$$

因此，我们很容易得到小波子空间W_j的各种分解如下：

$$\left.\begin{aligned} W_j &= U_{j-1}^2 \oplus U_{j-1}^3 \\ W_j &= U_{j-2}^4 \oplus U_{j-2}^5 \oplus U_{j-2}^6 \oplus U_{j-2}^7 \\ &\cdots \\ W_j &= U_{j-k}^{2^k} \oplus U_{j-k}^{2^k-1} \oplus \cdots \oplus U_{j-k}^{2^{k+1}} \oplus U_{j-k}^{2^{k-1}} \\ &\cdots \\ W_j &= U_0^{2^j} \oplus U_0^{2^j+1} \oplus \cdots \oplus U_0^{2^{j+1}-1} \end{aligned}\right\} \tag{1.57}$$

W_j空间分解的子空间序列可写作$U_{j-1}^{2^l-m}$，$m=0,1,\cdots, 2^l-1$；$l=1,2,\cdots,j$；$j=1,2,\cdots$。子空间序列$U_{j-1}^{2^l+m}$的标准正交基为$\{2^{-(j-1)/2}u_{2^l+m}(2^{j-l}t-k); k\in Z\}$。容易看出，当$l=0$和$m=0$时，子空间序列$U_{j-1}^{2^l+m}$简化为$U_j^1=W_j$，相应的正交基简化为$2^{-j/2}u_1(2^{-j}t-k)=2^{-j/2}\Psi(2^{-j}t-k)$，它恰好是标准正交小波族$\{\Psi_{j,k}(t)\}$。

若n是一个倍频程细划的参数，即令$n=2^l+m$，则我们有小波包的简略记号$\Psi_{j,k,n}(t)=2^{-j/2}\Psi_n(2^{-j}t-k)$，其中，$\Psi_n(t)=2^{l/2}u_{2^l+m}(2^l t)$。我们把$\Psi_{j,k,n}(t)$称为具有尺度指标$j$、位置指标$k$和频率指标$n$的小波包。将它与前面的小波$\Psi_{j,k}(t)$作一比较知，小波只有离散尺度$j$和离散平移$k$两个参数，而小波包除了这两个离散参数外，还增加了一个频率参数$n=2^l+m$。正是这个频率新参数的作用，使得小波包克服了小波时间分辨率高时频率分辨率低的缺陷。于是，参数n表示$\Psi_n(t)=2^{l/2}u_{2^l+m}(2^l t)$函数的零交叉数目，也就是其波形的振荡次数。

定义 1.8(小波库) 由$\Psi_n(t)$生成的函数族$\Psi_{j,k,n}(t)$(其中$n\in Z_+$；$j,k\in Z$)称为由尺度函数$\phi(t)$构造的小波库。

推论 1.1 对于每个$j=0, 1, 2, \cdots$

$$L^2(R) = \bigoplus_{j\in Z} W_j = \cdots \oplus W_{-1} \oplus W_0 \oplus U_0^2 \oplus U_0^3 \oplus \cdots \tag{1.58}$$

这时，族

$$\{u_{j,k}, u_n(t-k) \mid j=\cdots, -1, 0;\ n=2, 3, \cdots \text{ 且 } k\in Z\} \tag{1.59}$$

是$L^2(R)$的一个正交基。

随着尺度j的增大，相应正交小波基函数的空间分辨率愈高，而其频率分辨率愈低，

这正是正交小波基的一大缺陷。而小波包却具有将随 j 增大而变宽的频谱窗口进一步分割变细的优良性质，从而克服了正交小波变换的不足。

小波包可以对 W_j 进一步分解，从而提高频率分辨率，是一种比多分辨分析更加精细的分解方法，具有更好的时频特性。

1.6.4　小波包算法

下面给出小波包的分解算法和重构算法。设 $g_j^n(t) \in U_j^n$，则 g_j^n 可表示为

$$g_j^n(t) = \sum_l d_l^{j,n} u_n(2^j t - l) \tag{1.60}$$

小波包分解算法　由 $\{d_l^{j+1,n}\}$ 求 $\{d_l^{j,2n}\}$ 与 $\{d_l^{j,2n-1}$

$$\left.\begin{aligned} d_l^{j,2n} &= \sum_k a_{k-2l} d_k^{j+1,n} \\ d_l^{j,2n-1} &= \sum_k b_{k-2l} d_k^{j+1,n} \end{aligned}\right\} \tag{1.61}$$

小波包重构算法　由 $\{d_l^{j,2n}\}$ 与 $\{d_l^{j,2n+1}\}$ 求 $\{d_l^{j+1,n}\}$

$$d_l^{j+1,n} = \sum_k \left[h_{l-2k} d_k^{j,2n} + g_{l-2k} d_k^{j,2n+1}\right] \tag{1.62}$$

第 2 章 小波分析工具箱函数

在这一章里，我们将详细介绍在 MATLAB 中用于小波分析的各种工具箱函数，对每个函数从算法功能、语法格式、使用说明以及用法举例等方面进行阐述。

2.1 小波分析中的通用函数

1. biorfilt

功能：双正交小波滤波器组

格式：① [Lo_D, Hi_D, Lo_R, Hi_R]=biorfilt(DF, RF)

② [Lo_D1, Hi_D1, Lo_R1, Hi_R1, Lo_D2, Hi_D2, Lo_R2, Hi_R2]
= biorfilt(DF, RF,'8')

说明：该函数可以产生与双正交小波相关联的四个或八个滤波器组，DF 为分解滤波器，RF 为重构滤波器。格式①用来计算与由 DF，RF 所指定的双正交小波相关联的四个滤波器，Lo_D 为分解低通滤波器，Hi_D 为分解高通滤波器，Lo_R 为重构低通滤波器，Hi_R 为重构高通滤波器。格式②返回八个滤波器，前四个与分解小波相关，后四个与重构小波相关。

我们知道，对于一个子带滤波器，由于它们不满足严格对称和精确重构的条件，所以一个 FIR 滤波器在构造正交小波时存在一些不可避免的缺点。因此，我们就必须对小波的正交性作出让步，改用双正交小波，即

用 $\tilde{\Psi}$ 小波进行分解，对于一个信号 s，其分解系数为

$$\tilde{c}_{j,k} = \int s(x)\tilde{\Psi}_{j,k}(x)\mathrm{d}x$$

用 Ψ 小波进行重构，即

$$S = \sum_{j,k} \tilde{c}_{j,k}\Psi_{j,k}$$

两个小波之间存在着如下的关系：

$$\int \tilde{\Psi}_{j,k}(x)\Psi_{j',k'}(x)\mathrm{d}x = 0 \qquad j \neq j';\ k \neq k'$$

$$\int \phi_{0,k}(x)\phi_{0,k'}(x)\mathrm{d}x = 0 \qquad k \neq k'$$

举例：
```
%计算与 bior3.5 相关联分解滤波器 DF 和重构滤波器 RF
[RF, DF]=biorwavf('bior3.5');
%计算所需的四个滤波器
[Lo_D, Hi_D, Lo_R, Hi_R]=biorfilt(DF, RF);
subplot(221); stem(Lo_D);
```

```
title('分解低通滤波器'); grid;
subplot(222); stem(Hi_D);
title('分解高通滤波器'); grid;
subplot(223); stem(Lo_R);
title('重构低通滤波器'); grid;
subplot(224); stem(Hi_R);
title('重构高通滤波器'); grid;
```

输出结果(如图 2.1 所示):

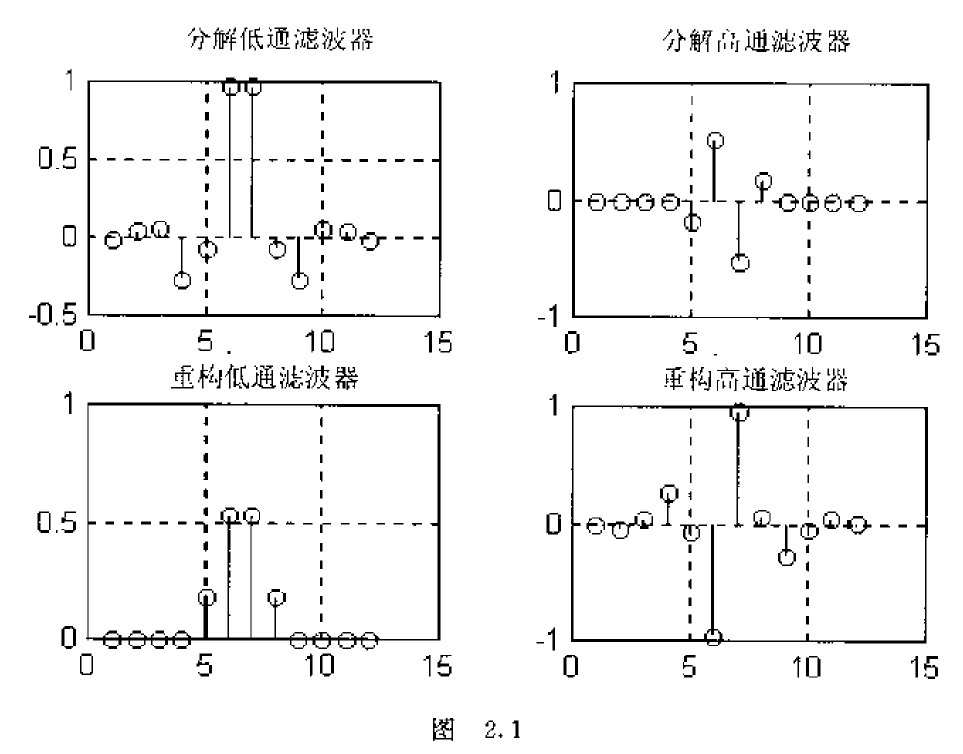


图　2.1

参见: biorwavf, orthfilt

2. dyaddown

功能: 二元取样

格式: ① Y=dyaddown(X,evenodd)

② Y=dyaddown(X)

③ Y=dyaddown(X,evenodd,'type')

④ Y=dyaddown(X,'type', evenodd)

说明: 该函数是用来从向量 X 中每隔一个元素抽取一个元素组成向量 Y。对于格式①，如果 evenodd 为偶数，则进行偶取样，即 Y(k)=X(2k)；如果 evenodd 为奇数，则进行奇取样，即 Y(k)=X(2k+1)，如果省略 evenodd 的值(即格式②)，则 evenodd=0,Y(k)=X(2k)。对于格式③④的情况，X 是一个矩阵，如果 type=c，则抽取矩阵 X 偶(或奇)数列，如果 type=r，则抽取矩阵 X 的偶(或奇)数行，如果 type=m,则抽取矩阵 X 的偶(或奇)数行和偶(或奇)数列。

举例：s=1：10 %定义一个元素由1到10的向量s

```
dse=dyaddown(s) %进行偶数序号抽取
%dse=dyaddown(s,0) %进行偶数序号抽取
dso=dyaddown(s,1) %进行奇数序号抽取
a=[1,2,3,4;5,6,7,8;9,10,11,12;13,14,15,16] %定义一个矩阵a
dar=dyaddown(a,0,'r') %抽取矩阵a的偶数行
dac=dyaddown(a,1,'c') %抽取矩阵a的奇数列
dam=dyaddown(a,0,'m') %抽取矩阵a的偶数行和偶数列
```

输出结果：

```
s =
     1     2     3     4     5     6     7     8     9    10
dse =
     2     4     6     8    10
dso =
     1     3     5     7     9
a =
     1     2     3     4
     5     6     7     8
     9    10    11    12
    13    14    15    16
dar =
     5     6     7     8
    13    14    15    16
dac =
     1     3
     5     7
     9    11
    13    15
dam =
     6     8
    14    16
```

参见：dyaddup

3. dyadup

功能：二元插值

格式：① Y=dyadup(X, evenodd)

② Y=dyadup(X)

③ Y=dyadup(X, evenodd,'type')

④ Y=dyadup(X, 'type',evenodd)

说明：该函数是用来从向量 X 中每隔一个元素填充一个 0 元素组成向量 Y，该函数在小波重构算法中非常有用。对于格式①，如果 evenodd 为偶数，则进行偶插值，即 Y(2k−1)＝X(k)，Y(2k)＝0；如果 evenodd 为奇数，则进行奇插值，即 Y(2k−1)＝0，Y(2k)＝X(k)；如果省略 evenodd 的值，则 evenodd＝0，Y(2k−1)＝X(k)，Y(2k)＝0。对于格式③④的情况，X 是一个矩阵，如果 type＝c，则插值矩阵 X 偶(或奇)数列；如果 type＝r，则插值矩阵 X 的偶(或奇)数行；如果 type＝m，则插值矩阵 X 的偶(或奇)数行和偶(或奇)数列。

举例：
```
s=1:5; %定义一个元素由 1 到 10 的向量 s
use=dyadup(s,0) %进行偶数序号插值
uso=dyadup(s,1) %进行奇数序号插值
a=[1,2;3,4;5,6;7,8] %定义一个矩阵 a
uar=dyadup(a,0,'r') %插值矩阵 a 的偶数行
uac=dyadup(a,1,'c') %插值矩阵 a 的奇数列
uam=dyadup(a,0,'m') %插值矩阵 a 的偶数行和偶数列
```

输出结果：
```
s =
     1     2     3     4     5
use =
     1     0     2     0     3     0     4     0     5
uso =
     0     1     0     2     0     3     0     4     0     5     0
a =
     1     2
     3     4
     5     6
     7     8
uar =
     1     2
     0     0
     3     4
     0     0
     5     6
     0     0
     7     8
uac =
     0     1     0     2     0
     0     3     0     4     0
     0     5     0     6     0
     0     7     0     8     0
```

```
uam =
      1    0    2
      0    0    0
      3    0    4
      0    0    0
      5    0    6
      0    0    0
      7    0    8
```

参见：dyaddown

4. wavefun

功能：小波函数和尺度函数

格式：① [phi,psi,Xval]=wavefun('wname',iter)

　　② [phi1,hsi1,phi2,pis2,Xval]=wavefun('wname',iter)

　　③ [psi,Xval]=wavefun('wname',iter)

　　④ wavefun('wname',a,b)

说明：该函数用来返回小波函数 Ψ 和相应的尺度函数 ϕ(在尺度函数存在的情况下)的近似值。正整数 iter 决定了反复计算的次数，从而确定了近似值的精确程度。

- 对于一个正交小波，格式①返回尺度函数和小波函数，X 在支撑区间上有 2^{iter} 个点。
- 对于一个双正交小波，格式②返回分别用于分解的尺度函数($\phi1$)和小波函数($\Psi1$)及重构的尺度函数($\phi2$)和小波函数($\Psi2$)。
- 对于一个 Meyer 小波，有：[phi,psi,Xval]=wavefun('wname',iter)。
- 对于一个 Morlet 小波或 Mexican Hat 小波，有 [psi,Xval]=wavefun('wname',iter)。

对于格式④，a,b 是正整数，且格式④等价于 wavefun('wname',max(a,b))。它计算尺度函数和小波函数的近似值并画出图形。

举例：

```
iter=10; wav='sym4'; %设置小波的名字和计算的次数
%下面用叠代算法计算小波函数Ψ的近似值并画出波形图
for I=1:iter
    [phi,psi,Xval]=wavefun(wav,I);
    plot(Xval,psi);
    hold on
end
title('小波函数 sym4 的近似值(iter 从 1 到 10)');
hold off
```

输出结果(如图 2.2 所示)。

参见：intwave，waveinfo，wfilters

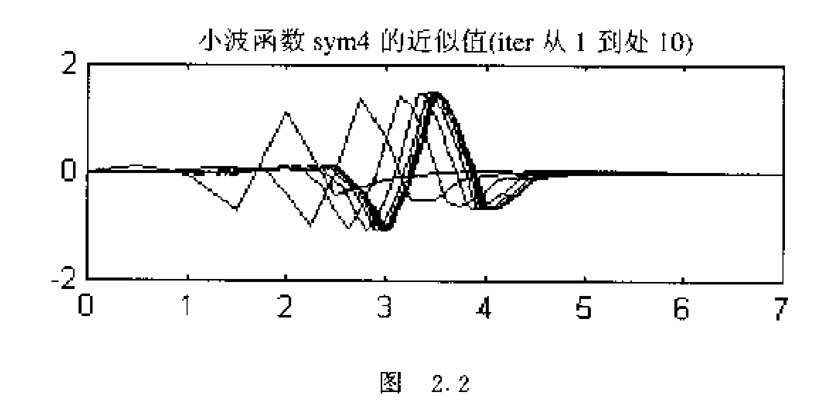


图　2.2

5. intwave

功能：积分小波函数 Ψ

格式：① [Integ,Xval]=intwave('wname',Prec)

② [Integ,Xval]=intwave('wname',Prec,Pflag)

③ [Integ,Xval]=intwave('wname')

说明：该函数返回小波函数 Ψ 的积分值，积分区间为[−∞,x]，即返回值 $\int_{-\infty}^{x}\Psi(y)\mathrm{d}y$。当采用三个参数的格式时，第三个参数为虚设参数，如果 Pflag 不为 0，则格式②画出在 x 上不为 0 的积分值。

另外，[Integ,Xval]=intwave('wname',Prec)等价于 [Integ,Xval]=intwave('wname', Prec, 0)；

[Integ,Xval]=intwave('wname')等价于 [Integ,Xval]=intwave('wname', 8)。

intwave 仅仅用于连续小波分析。

举例：
```
wname='db4'; %把小波 db4 赋给 wname
%下面画出小波函数的图形
[phi,psi,xval]=wavefun(wname,7);
subplot(321); plot(xval,psi);
title('小波');
%计算并画出小波积分的近似值
[integ,xval]=intwave(wname,7);
subplot(322); plot(xval,integ);
title('对于每个 x，小波在[−inf,x]上的积分');
```
输出结果(如图 2.3 所示)。

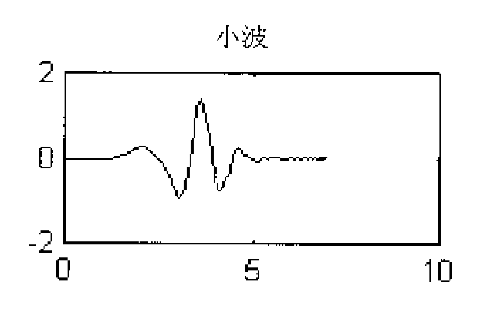
参见：wavefun

6. orthfilt

功能：正交小波滤波器组

格式：① [Lo_D,Hi_D,Lo_R,Hi_R]=orthfilt(W)

说明：该函数用来计算与某一小波对应的尺度滤波器相关的四个滤波器，其中，Lo_D 为分解的低通滤波器，Hi_D 为分解的高通滤波器，Lo_R 为重构的低通滤波器，Hi_R 为重



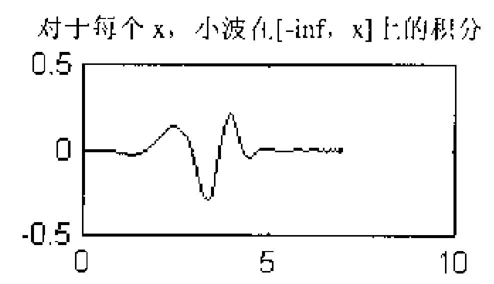


图　2.3

构的高通滤波器。

对于一个正交小波，在多分辨分析应用中，我们是用尺度函数 ϕ 和小波函数 Ψ 进行分析的，且尺度函数存在着一个基本的双尺度关系

$$\frac{1}{2}\phi\left(\frac{x}{2}\right) = \sum_{n\in Z} w_n\phi(x-n)$$

在 dwt 和 idwt 中，所有的滤波器都与序列 $(w_n)_{n\in Z}$ 密切相关。很清楚，如果 ϕ 具有紧支撑，则序列是有限的，且可以看作是一个 FIR 滤波器。尺度滤波器 W 为：(1)低通 FIR 滤波器；(2)长度为 2N；(3)总和为 1；(4)方差为 $1/\sqrt{2}$。

对于滤波器 W，我们可以定义四个 FIR 滤波器，其长度为 2N，方差为 1，具体见表 2－1。

表 2－1　滤 波 器 类 型

滤波器	低通	高通
分解滤波器	Lo_D	Hi_D
重构滤波器	Lo_R	Hi_R

四个滤波器在计算上的转换关系如图 2.4 所示。

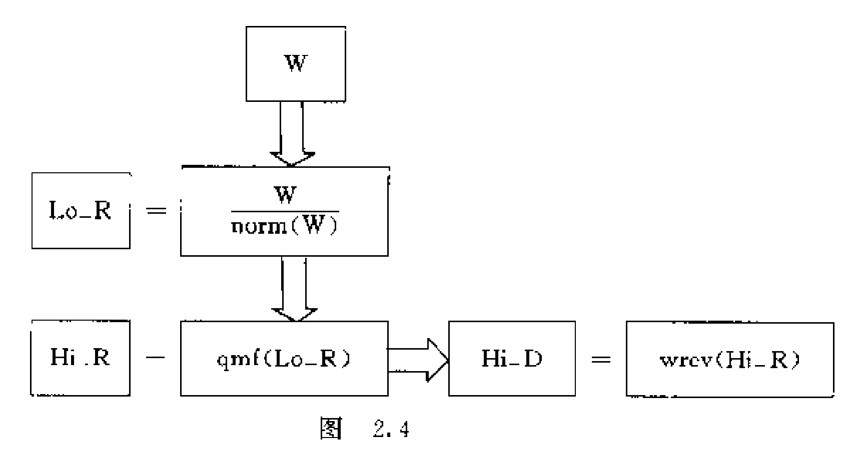


图　2.4

注意：qmf 使得 Hi_R 和 Lo_R 为镜像二次滤波器，wrev 可以回转滤波器系数，所以 Hi_D 和 Lo_D 也为镜像二次滤波器。

举例：load db8；w=db8；%给 w 装载一个尺度滤波器

subplot(421)；stem(w)；grid；

```
title('原始尺度滤波器');
%下面计算四个滤波器
[Lo_D,Hi_D,Lo_R,Hi_R]=orthfilt(w);
subplot(423); stem(Lo_D);
title('分解的低通滤波器'); grid;
subplot(424); stem(Hi_D);
title('分解的高通滤波器'); grid;
subplot(425); stem(Lo_R);
title('重构的低通滤波器'); grid;
subplot(426); stem(Hi_R);
title('重构的高通滤波器'); grid;
%检查双正交性
df=[Lo_D;Hi_D];
rf=[Lo_R;Hi_R];
disp('检查双正交性的结果, df * rf=:');
id=df * df'
%高频和低频图例说明
fftld=fft(Lo_D); ffthd=fft(Hi_D);
freq=[1:length(Lo_D)]/length(Lo_D);
subplot(427);plot(freq,abs(fftld));
title('转换模量: 低通'); grid;
subplot(428); plot(freq,abs(ffthd));
title('转换模量: 高通'); grid;
```

输出结果(如图 2.5 所示)。

```
检查双正交性的结果, df * rf=:
id =
    1.0000         0
         0    1.0000
```

参见: biorfilt, qmf, wfilters

7. qmf

功能: 镜像二次滤波器(quadrature mirror filters,简称 QMF)

格式: ① Y=qmf(X,P)

② Y=qmf(X)

说明: 对于格式①，当 P 为偶数时，函数改变向量 X 中偶数位置的元素的符号；当 P 为奇数时，函数改变向量 X 中奇数位置的元素的符号。Y=qmf(X)等价于 Y=qmf(X, 0)。

若 X 是一个能量有限的信号，如果对任意 X 满足(即 QMF 条件):

$$\|\gamma_0\|^2 + \|\gamma_1\|^2 = \|x\|^2$$

则对应的滤波器 F_0 和 F_1 为镜像二次滤波器。

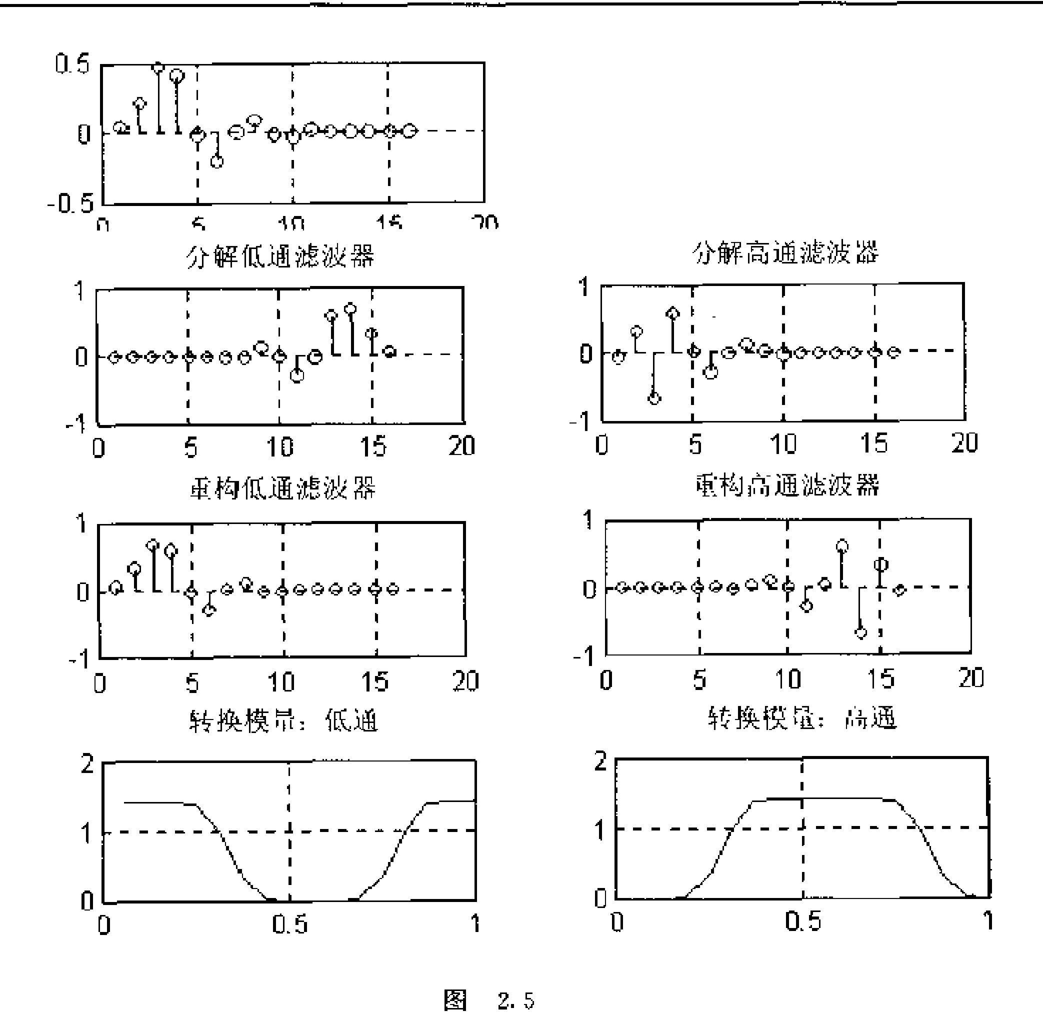


图　2.5

例如，F_0 是一个 Daubechies 尺度滤波器且 $F_1 = \mathrm{qmf}(F_0)$，则波滤器 F_0 和 F_1 的转换函数满足：

$$|F_0(z)|^2 + |F_1(z)|^2 = 1$$

举例：%装载与某一正交小波相关的尺度滤波器

```
load db10;
subplot(621); stem(dh10); title('db10 低通滤波器'); grid;
%计算积分镜像滤波器
qmfdb10=qmf(db10);
subplot(622); stem(qmfdb10); title('QMF 低通滤波器'); grid;
%检查频率条件
m=fft(db10);
mt=fft(qmfdb10);
freq=[1:length(db10)]/length(dh10);
subplot(625); plot(freq,abs(m));
title('db10 的转换模式'); grid;
subplot(626);plot(freq,abs(mt));
title('QMF db10 的转换模式'); grid;
```

```
subplot(629); plot(freq,abs(m).^2+abs(mt).^2);
title('检查 db10 和 QMF db10 的 QMF 条件'); grid;
xlabel('abs(fft(db10))^2+abs(fft(qmf(db10))^2=1');
%检查正交性
df=[db10;qmfdb10]*sqrt(2);
id=df*df'
```

输出结果(如图 2.6)所示:

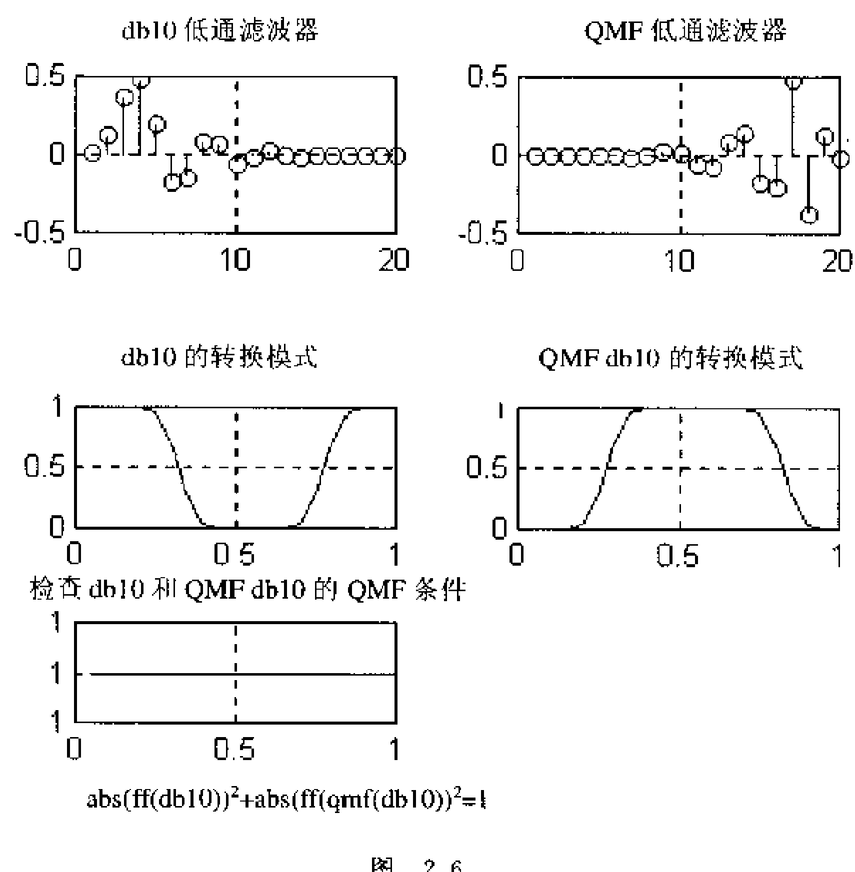


图　2.6

```
id =
      1.0000      0.0000
      0.0         1.0000
```

参见: waveinfo

8. wfiters

功能: 小波滤波器

格式: ① [Lo_D, Hi_D, Lo_R, Hi_R]=wfilters('wname')

　　② [F1, F2]=wfilters('wname','type')

说明: 格式①是用来计算和正交或双正交小波 wname 相关联的四个滤波器。返回的四个滤波器分别是 Lo_D (分解的低通滤波器)、Hi_D(分解的高通滤波器)、Lo_R(重构的低通滤波器)、Hi_R(重构的高通滤波器)。Lo_D 和 Hi_D 分别是 Lo_R 和 Hi_R 的对偶算子，也

可以分别理解为Lo_R和Hi_R的共轭转置矩阵。在信号处理中，它们的作用可以用图2.7表示(以多分辨分析为例说明)。

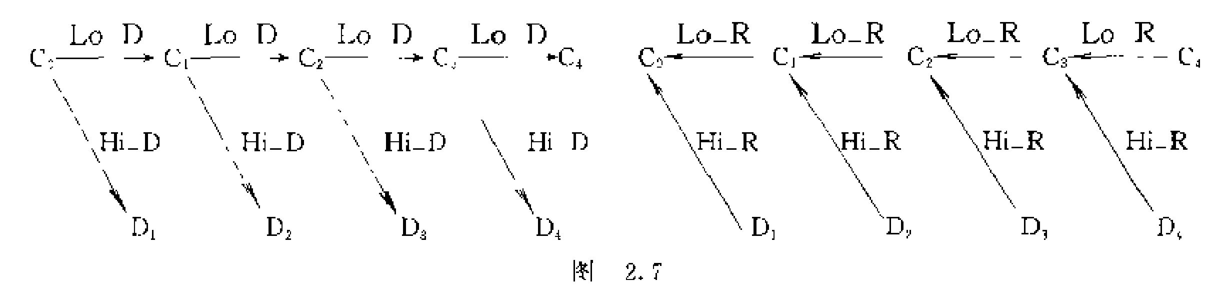


图 2.7

可用到的正交或双正交小波函数 wname 有：

Daubechies ：db1 或 haar，db2，db3，… ,db10，… ,db50

Coiflets ：coif1，… ,coif5

Symlets ：sym2，… ,sym8

Biorthogonal ：bior1.1，bior1.3，bior1.5
bior2.2，bior2.4，bior2.6，bior2.8
bior3.1，bior3.3，bior3.5，bior3.7
bior3.9，bior4.4，bior5.5，bior6.8

调用格式②返回如下两个滤波器：

Lo_D 和 Hi_D (分解滤波器)　　type 为'd'时

Lo_R 和 Hi_R (重构滤波器)　　type 为'r'时

Lo_D 和 Hi_R (低通滤波器)　　type 为'l'时

Lo_D 和 Hi_R (高通滤波器)　　type 为'h'时

举例：

```
wname='db5'; %wname 取 db5 小波
%下面计算与给定小波相关联的四个滤波器
[Lo_D, Hi_D, Lo_R, Hi_R]=wfilters(wname);
subplot(421); stem(Lo_D); title('分解低通滤波器'); grid;
subplot(422); stem(Hi_D); title('分解高通滤波器'); grid;
subplot(425); stem(Lo_R); title('重构低通滤波器'); grid;
subplot(426); stem(Hi_R); title('重构高通滤波器'); grid;
```

输出结果(如图2.8所示)。

参见：biorfilt，orthfilt，waveinfo

9. wavemngr

功能：小波管理

格式：

① wavemngr('create')

② wavemngr('add',Fn,Fsn,Wt,Nums,File)

③ wavemngr('add',Fn,Fsn,Wt,Nums,File,B)

④ wavemngr('del',N)

⑤ wavemngr('restore')

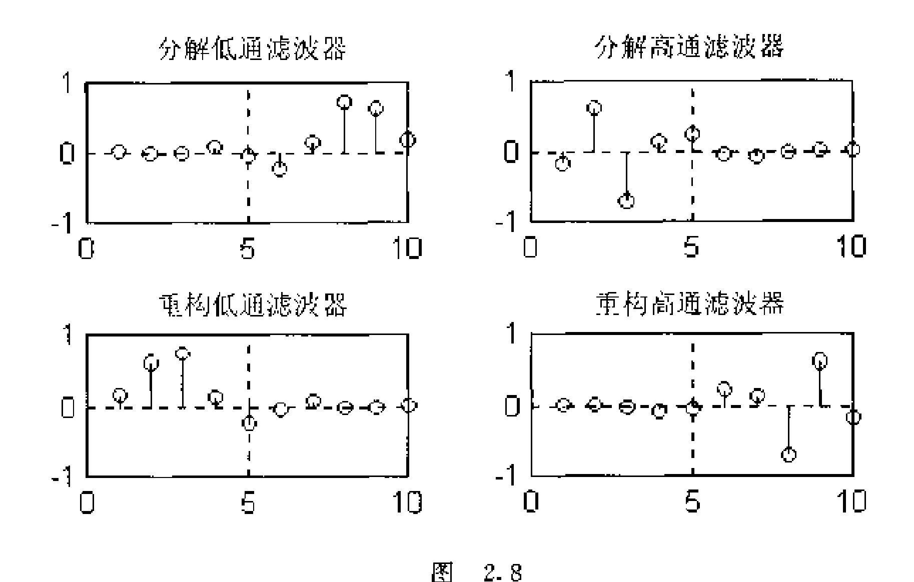


图　2.8

⑥ wavemngr('restroe',In2)

⑦ Out1=wavemngr('read')

⑧ Out1=wavemngr('read',In2')

⑨ Out1=wavemngr('read_asc')

说明：该函数是一个用于小波管理的函数，它可以对小波进行新建、添加、删除、存储和读取操作。对于格式①，该函数可以用来创造一个新的小波函数(扩展名为 *.inf 或 *.asc)；对于②、③，该函数可以把一个新的小波函数加到工具箱中，Fn 是 Family Name 的缩写，Fsn 是 Family Short Name 的缩写，Wt 是 Wavelet Type 的缩写。当 Wt=1 时，为正交小波，当 Wt=2 时，为双正交小波，当 Wt=3 时，为带尺度函数的小波，当 Wt=4 时，为不带尺度函数的小波。Nums 为字符串的个数，File 为小波函数名，对于 Wt 为 3、4 的情况，B 为小波有效支撑的上下边界；对于④，该函数是用来删除一小波，N 为小波函数名；对于⑤、⑥，该函数用来保存先前的小波函数；对于⑦、⑧、⑨，该函数是用来读取小波函数。

举例：%下面列出在 MATLAB 中已有的小波函数

```
wavemngr('read')
```

输出结果：

```
ans =
=================================
Haar              haar
Daubechies        db
BiorSplines       bior
Coiflets          coif
Symlets           sym
Morlet            morl
Mexican_hat       mexh
Meyer             meyr
=================================
```

```
%下面列出在MATLAB中存在的所有小波函数
wavemngr('read',1)
```

输出结果：

```
ans =
===========================
Haar                haar
===========================
Daubechies          db
--------------------
db1 db2 db3 db4
db5 db6 db7 db8
db9 db10 db * *
===========================
BiorSplines         bior
--------------------
bior1.1    bior1.3    bior1.5    bior2.2
bior2.4    bior2.6    bior2.8    bior3.1
bior3.3    bior3.5    bior3.7    bior3.9
bior4.4    bior5.5    bior6.8
===========================
Coiflets         coif
--------------------
coif1 coif2 coif3 coif4
coif5
===========================
Symlets           sym
--------------------
sym2 sym3 sym4 sym5
sym6 sym7 sym8
===========================
Morlet              morl
===========================
Mexican  hat        mexh
===========================
Meyer               meyr
===========================
```

在下面的例子中，我们将一个具有紧支撑的正交小波添加到工具箱中，这些小波是在运用伯恩斯坦(Bernstein)多项式的基础上对Daubechies小波的推广。在添加一个正交小波时，必须要进行如下的定义：

小波组名：如 Lemarie

小波组缩写名：如 lem

小波类型：如 1(orth)

小波数目：如 1 2 3 4 5

文件驱动：如 lemwavf

程序：

```
wavemngr('add','Lemarie','lem',1,'1 2 3 4 5','lemwavf');
wavemngr('read')
```

输出结果：

```
ans =
==============================
  Haar                haar
  Daubechies          db
  BiorSplines         bior
  Coiflets            coif
  Symlets             sym
  Morlet              morl
  Mexican_hat         mexh
  Meyer               meyr
  Lemarie             lem
==============================
```

```
%把该文件从小波组中删除
wavemngr('del','Lemarie');
wavemngr('read')
```

输出结果：

```
ans =
==============================
  Haar                haar
  Daubechies          dh
  BiorSplines         bior
  Coiflets            coif
  Symlets             sym
  Morlet              morl
  Mexican_hat         mexh
  Meyer               meyr
==============================
```

```
%下面存储并显示小波文件
wavemngr('restore');
wavemngr('read',1);
```

输出结果：

```
ans =
===========================
Haar                haar
===========================
Daubechies          db
--------------------
db1 db2 db3 db4
db5 db6 db7 db8
db9 db10 db * *
===========================
BiorSplines         bior
--------------------
bior1.1     bior1.3     bior1.5     bior2.2
bior2.4     bior2.6     bior2.8     bior3.1
bior3.3     bior3.5     bior3.7     bior3.9
bior4.4     bior5.5     bior6.8
===========================
Coiflets        coif
--------------------
coif1 coif2 coif3 coif4
coif5
===========================
Symlets             sym
--------------------
sym2 sym3 sym4 sym5
sym6 sym7 sym8
===========================
Morlet              morl
===========================
Mexican  hat        mexh
===========================
Meyer               meyr
===========================
Lemarie             lem
--------------------
lem1 lem2 lem3 lem4
lem5
===========================
```

此时，该函数便存入小波组中，在命令行或GUI中可进行调用。但在使用该函数时必须注意一点，所添加的小波函数必须是一个真正的小波函数，因为MATLAB不对此进行

检查。

10. wmaxlev

功能：计算小波分解的最大尺度

格式：L＝wmaxlev(S,'wname')

说明：wmaxlev 是一维或二维小波函数或小波包的导向函数，它可帮助你避开那些不必要的最大尺度值。该函数返回信号或图像的最大分解尺度。

举例：%对于一维信号的长度为 2^10 的情况

```
s=2^10;
w='db1'; l1=wmaxlev(s,w) %小波函数为 db1 时，计算分解的最大尺度值
w='db7'; l2=wmaxlev(s,w) %小波函数为 db7 时，计算分解的最大尺度值
```

输出结果：

```
l1 =
    10
l2 =
    6
```

```
%对于二维图像大小为[2^9,2^7]的情况
s=[2^9,2^7];
w='db1'; l1=wmaxlev(s,w) %小波函数为 db1 时，计算分解的最大尺度值
w='db7'; l2=wmaxlev(s,w) %小波函数为 db7 时，计算分解的最大尺度值
```

输出结果：

```
l1 =
    7
l2 =
    3
```

参见：wavedec，wavedec2，wpdec，wpdec2

11. deblankl

功能：把字符串变成无空格的小写字符串

格式：① S＝deblankl(X)

说明：该函数是一个通用的函数，它可以对字符串进行灵活的运用，把一般的字符串变成一个无空格的小写字符串。

举例：x='AB1 C %9'

```
y=deblankl(x)
```

输出结果：

```
x =
AB1 C %9
y =
ab1c%9
```

12. errargn

功能：检查函数参数数目

格式：① err＝errargn('function',numargin,argin,numargout,argout)

说明：该函数是一个通用函数。Function 是被检查的函数，numargin 是输入参数的数目，argin 是输入参数，numargout 是输出参数的数目，argout 是输出参数。如果指定函数的输入参数数目和输出参数数目分别与 numargin、numargout 相等，则返回参数 err＝1，并输出错误信息；如果指定函数的输入参数数目和输出参数数目分别与 numargin、numargout 不相等，则返回参数 err＝0。

举例：err1＝errargn('line',4,[2,3],0,[0,1])

　　err2＝errargn('surf',4,[3,4],0,[0,1])

输出结果：

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
ERROR …
------------------------------
line ---> invalid number of arguments
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
err1 =
     1
err2 =
     0
```

参见：errargt

13. errargt

功能：检查函数的参数类型

格式：① ERR＝errargt(DNFCT,VAR,TYPE)

　　② ERR＝errargt(NDFCT,VAR,'msg')

说明：该函数是一个通用的函数，如果输入向量或矩阵的元素的类型与指定的类型一致，则返回 0，否则返回 1，并在命令窗口中显示错误信息。

TYPE 所具有的类型如下：

int —— 严格的正整数(不包括 0)

in0 —— 正整数(包括 0)

rel —— 整数

rep —— 严格的正实数(不包括 0)

re0 —— 正实数(包括 0)

str —— 字符串

vec —— 向量

row —— 行向量

col —— 列向量

dat —— 日期，日期的表示形式为 AAAAMMJJHHMNSS，其中 AAAA 表示年，且有 0≤AAAA≤9999；MM 表示月，且有 0≤MM≤12；JJ 表示日，且有 0≤JJ≤31；HH 表示小时，且有 0≤HH≤23；MN 表示分钟，且有 0≤MN≤59；SS 表示秒，且有0≤SS≤59。

mon —— 月份，月份的表示形式为 0≤MN≤12。

参见：errargn

14. num2mstr

功能：把数字转化为字符串

格式：S=num2mstr(N)

说明：该函数用来把一个实数矩阵 N 转化成一个字符串向量 S。

举例：
```
N=[1.32451,3.24354;5.23454,6.23454];
S=num2mstr(N)
```
输出结果：
```
N =
    1.3245    3.2435
    5.2345    6.2345
S =
[1.32451 3.24354; 5.23454 6.23454;]
```

参见：num2str

15. wcodemat

功能：对矩阵进行量化编码

格式：① Y=wcodemat(X,NB,OPT,ABSOL)

② Y=wcodemat(X,NB,OPT)

③ Y=wcodemat(X,NB)

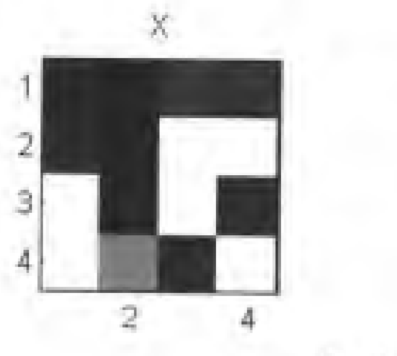
④ Y=wcodemat(X)

说明：该函数是用来对矩阵 X 进行量化编码，它返回矩阵 X 的一个编码矩阵，在编码中，把矩阵 X 中元素绝对值最大的作为 NB(NB 是一个整数)，绝对值最小的作为 1，其它元素依其绝对值的大小在 1 与 NB 中排列。当 OPT 为'row'时，编码进行行编码，即把每行元素中绝对值最大的元素作为 NB，最小的作为 1，其它元素依其绝对值的大小在该行中进行排列；当 OPT 为'col'时，编码进行列编码，即把每列元素中绝对值最大的元素作为 NB，最小的作为 1，其它元素依其绝对值的大小在该列中进行排列；当 OPT 为'mat'时，编码全局编码，即把整个矩阵中元素中绝对值最大的元素作为 NB，最小的作为 1，其它元素依其绝对值的大小在整个矩阵中进行排列。当 ABSOL 为 0 时，该函数返回输入矩阵 X 的一个编码版本(coded version)，当 ABSOL 为非 0 值时，该函数返回输入矩阵 X 的 ABS(X)。

举例：
```
%输出入一个矩阵
X=[23.245,-234.342,54.234,34.235;
    35.236,-64.465,423.243,243.123;
    443.255,-121.354,546.232,24.242;
```

```
    245.234,143.234,24.234,234.255];
%下面对X矩阵进行量化编码
Y=wcodemat(X,44,'mat',8)
%画出原始矩阵的图像
subplot(221); image(X); colormap(map);
title('X');
axis square
%画出量化编码后的图像
subplot(222); image(Y); colormap(map);
title('Y');
axis square
```

输出结果(如图2.9所示)：



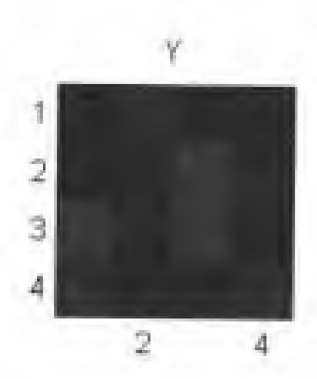


图　2.9

```
Y =
     1    18     3     1
     2     4    34    19
    36     9    44     1
    19    11     1    18
```

可以明显看出，经过量化编码后的图像比原来的图像在颜色的对比上要变得柔和多了。

16. wcommon

功能：寻找公共元素

格式：[XI, YI]=wcommon(X,Y)

说明：该函数用来寻找向量X与向量Y的公共元素，其返回值是由0、1元素组成的向量，即

当向量X中某一元素X(k)∈Y时，则XI|k|=1，否则等于0。

当向量Y中某一元素Y(k)∈X时，则YI|k|=1，否则等于0。

举例：%定义两个向量x,y

```
x=[10,20,30,40,50];
y=[60,50,70,30,20,12,31];
```

```
%寻找公共元素
[xi,yi]=wcommon(x,y)
%列出公共元素
comelem=x(find(xi))
```

输出结果：

```
xi =
    0    1    1    0    1
yi =
    0    1    0    1    1    0    0
comelem =
    20    30    50
```

参见：biorfilt，qmf，wfilters

17. wkeep

功能：提取向量或矩阵中的一部分

格式：① Y=wkeep(X，L，0)

② Y=wkeep(X，L)

说明：wkeep 是用来从向量 X 中提起一个向量 Y，L 为提取向量 Y 的长度。对于格式①，当 0='c'(即 center 的第一个字母)时，则 Y 是从 X 的中间提取，当 0='l'(即 left 的第一个字母)时，则 Y 是从 X 的左边提取，当 0='r'(即 right 的第一个字母)时，则 Y 是从 X 的右边提取；对于格式②则是默认从中间提取。

举例：

```
x=1：10；
y=wkeep(x，6，'c')；
```

输出结果：

```
y=
    3    4    5    6    7    8
    y=wkeep(x，6)
y=
    3    4    5    6    7    8
    y=wkeep(x,7,'l')
y=
    1    2    3    4    5    6    7
    y=wkeep(x,6,'r')
y=
    5    6    7    8    9    10
```

18. wrev

功能：向量逆序

格式：Y=wrev(X)

说明：　该函数得到原向量逆序排列的向量。

举例：%设置一个向量 x

```
x=[1,2,3];
%把行向量进行翻转
y1=wrev(x)
%把列向量进行翻转
y2=wrev(x')
```

输出结果：

```
y1 =
     3     2     1
y2 =
     3
     2
     1
```

参见：fliplr，flipud

19. nstdfft

功能：非标准一维快速博里叶变换(FFT)

格式：[XHAT，OMEGA]=nstdfft(X,LOWB,UPPB)

说明：该函数是一个通用的数学函数，它用来计算 X 在有效支撑[LOWB，UPPB](含 2^N 个样点)上的标准 FFT 变换。输出参数 XHAT 是时间间隔[−n:2:n−2]/(2 * (UPPB - LOWB))上的 FFT 变换值，其中，n 为 X 的长度，XHAT 是一个 n 维向量。

举例：%装入信号

```
load leleccum;
s=leleccum(1:1024);
%画出原始信号的波形
subplot(311); plot(s);
title('原始信号');
[xhat,omega]=nstdfft(s,1,1024);
%画出信号 nstdfft 变换后的频谱
subplot(312); plot(abs(xhat));
title('nstdfft 变换后的频谱');
%进行反变换并画出其波形
x=instdfft(xhat,1,1024);
subplot(313); plot(x);
title('instdfft 反变换后的信号');
```

输出结果(如图 2.10 所示)。

参见：wavedec，wavedec2，wpdec，wpdec2

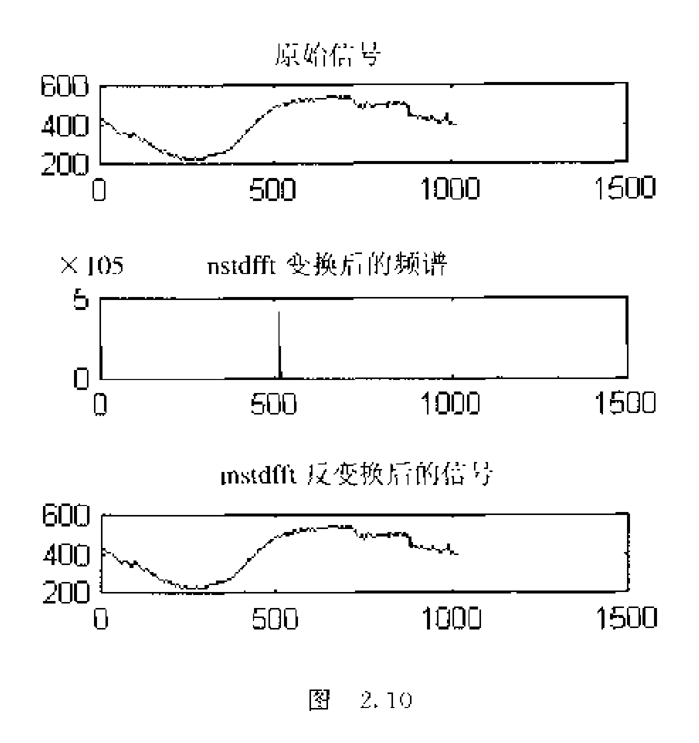


图　2.10

20. instdfft

功能：非标准一维快速逆傅里叶变换

格式：[X,T]=instdfft(XHAT,LOWB,UPPB)

说明：该函数是一个通用的数学函数，它用来计算 XHAT 在有效支撑[LOWB，UPPB]（含 2^N 个样点）上的非标准逆 FFT 变换。输出参数 X 是时间点 T 的恢复信号，其中，T=LOWB+[0：n－1]*(UPPB－LOWB)/n(n 为 XHAT 的长度)，它是一个 n 维向量。

参见：fft,ifft,nstdfft

21. std

功能：计算标准差

格式：① SNRAT=std(X)

② SNRAT=std(X,1)

③ SNRAT=std(X,FLAG,DIM)

说明：该函数用来计算向量或矩阵的标准差。如果 X 为向量，则返回向量的标准差；如果 X 为矩阵，则返回矩阵每列的标准差。

举例：
```
X1=[1,2,3,4,5];
X2=[1,2,3,4,5;6,7,8,9,10];
SX1=std(X1)
SX2=std(X2)
```

输出结果：
```
SX1 =
    1.5811
```

SX2 =

3.5355 3.5355 3.5355 3.5355 3.5355

2.2 小波函数

在这一节里，我们将介绍一个 MATLAB 工具箱中的小波函数。由于小波函数的定义方式有多种多样，有在时域中进行定义的，有在频域中进行定义的，有的则是通过滤波器进行定义的。根据不同小波函数的不同定义方式，可以计算出该小波函数的时域数值或滤波器数值。有关各小波函数的详细内容，请参见第 1.2 节。

1. biorwavf

功能：双正交样条小波滤波器

格式：[RF，DF]=biorwavf(W)

说明：该函数返回与指定双正交小波 W 相关联的两个尺度滤波器，W='biorNr.Nd'，其中 Nr 与 Nd 的可能情况如下：

Nr=1 Nd=1,3 或 5

Nr=2 Nd=2,4,6 或 8

Nr=3 Nd=1,3,5,7 或 9

Nr=4 Nd=4

Nr=5 Nd=5

Nr=6 Nd=8

输出参数是滤波器，其中，RF 为重构滤波器，DF 为分解滤波器。

举例：wname='bior2.2'；%设置正交样条小波名

[RF,DF]=biorwavf(wname) %计算与指定小波相关联的两个尺度滤波器

输出结果：

RF =

0 0.2500 0.5000 0.2500 0 0

DF =

−0.1250 0.2500 0.7500 0.2500 −0.1250 0

参见：biorfilt，waveinfo

2. coifwavf

功能：Coiflets 小波滤波器

格式：F=coifwavf(W)

说明：该函数返回与所指定的 coiflet 小波相关联的尺度滤波器，其中 coifN 中，N 的可能值有：1，2，3，4 或 5。

举例：wname='coif2'；%指定一个小波名

f=coifwavf(wname) %计算相应的尺度滤波器

输出结果：

```
f =
  Columns 1 through 7
      0.0116   -0.0293   -0.0476    0.2730    0.5747    0.2949   -0.0541
  Columns 8 through 12
     -0.0420    0.0167    0.0040   -0.0013   -0.0005
```

参见：waveinfo

3. dbaux

功能：Daubechies 小波滤波器

格式：① W=dbaux(N,SUMW)

② W=dbaux(N)

说明：格式①返回一个阶数为 N 的 Daubechies 尺度滤波器，N 可以取 1，2，…，sum(W)=SUMW。注意，当 N 的取值过大时，可能出现不稳定的情况。

另外，W=dbaux(N)等价于 W=dbaux(N,1)；W=dbaux(N,0)等价于 W=dbaux(N,1)。

举例：

```
%下面求与 db2 小波相对应的尺度滤波器
w=dbaux(2)
%下面对上面的结果进行验证
%db2 小波尺度滤波器 w 是方程 P=conv(wrev(w),w) * 2 的解
%P 是给定的对称滤波器 P=[-1/16,0,9/16,1,9/16,0,-1/16]
P=[-1/16,0,9/16,1,9/16,0,-1/16];
%上面方程的解 w 是基于 P 的根的最小相位解
rP=roots(P);
%保留在圆内的根，我们得到 db2 的滤波器为：
ww=poly([rP(6),-1,-1]);
ww=ww/sum(ww)
```

输出结果：

```
w =
    0.3415    0.5915    0.1585   -0.0915
ww =
    0.3415    0.5915    0.1585   -0.0915
```

从上面的例子可以看出，验证的结果与用 dbaux 函数求得的结果一样。

参见：dbwavf，wfilters

4. dbwavf

功能：Daubechies 小波滤波器

格式：F=dbwavf(W)

说明：该函数返回与所指定的 Daubechies 小波相关联的尺度滤波器，W='dbN'，阶数 N 的可能值为 1，2，3，…，50。

举例：wname='db4'；%指定小波函数名

f=dbwavf(wname)；%计算相应的尺度滤波器

输出结果：

```
f =
  Columns 1 through 7
    0.1629    0.5055    0.4461   -0.0198   -0.1323    0.0218    0.0233
  Column 8
   -0.0075
```

参见：dbaux，waveinfo，wfilters

5. mexihat

功能：墨西哥帽小波

格式：[PSI，X]=mexihat(LB,UB,N)

说明：该函数返回一个有效支撑为[LB,UB]，x 有 N 个均匀分布点的墨西哥帽小波，其中，返回向量 PSI 为 $\Psi(x)$的值，向量 X 为 x 点的取值。小波函数为

$$\Psi(x) = \left(\frac{2}{\sqrt{3}}\pi^{-1/4}\right)(1 - x^2)e^{-x^2/2}$$

它是高斯概率密度函数的二阶导数。

举例：%定义有效支撑的长度

```
lb=-5；ub=5；
%定义x在有效支撑上均匀点的个数
n=1000；
%下面计算并画出墨西哥帽小波
%其中，x为在有效支撑上均匀分布的点，psi为对应的Ψ(x)值
[psi,x]=mexihat(lb,ub,n)；
%画出小波的波形
subplot(221)；plot(x,psi)；
title('Mexihat 小波')；
```

输出结果(如图 2.11 所示)：

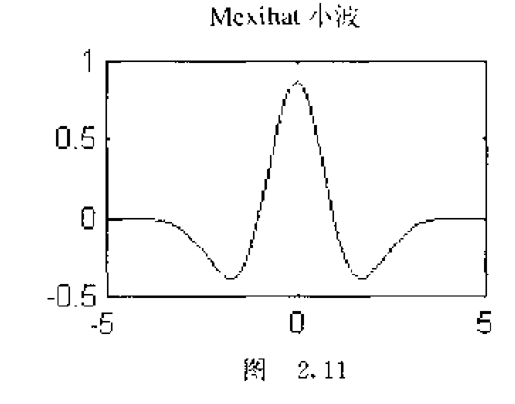


图 2.11

参见：waveinfo

6. meyer

功能：meyer 小波

格式：① [PHI,PSI,T]=meyer(LOWB,UPPB,N)

② [PHI,T]=meyer(LOWB,UPPB,N,'phi')

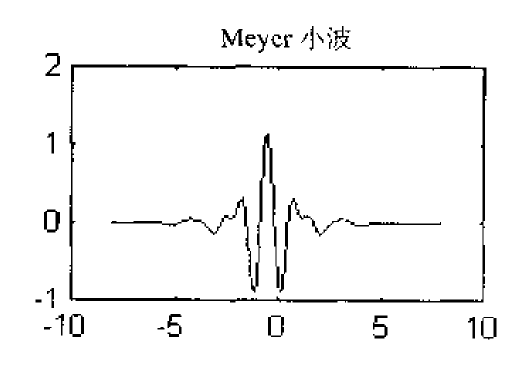
③ [PSI,T]=meyer(LOWB,UPPB,N,'psi')

说明：格式①返回 Meyer 小波在有效支撑为[LOWB,UPPB]，在有效支撑上有 N 个均匀分布点的 Meyer 尺度函数，N 为 2 的整数次幂。输出参数为尺度函数 phi(即 ϕ)和小波函数 psi(即 Ψ)。

举例：%定义小波函数有效支撑的长度

```
lowb=-8; uppb=8;
%定义在有效支撑上点的个数
n=1024;
%计算并画出 Meyer 小波和尺度函数
[phi,psi,x]=meyer(lowb,uppb,n);
subplot(221); plot(x,psi); title('Meyer 小波');
subplot(222); plot(x,phi); title('Meyer 尺度函数');
```

输出结果(如图 2.12 所示)：



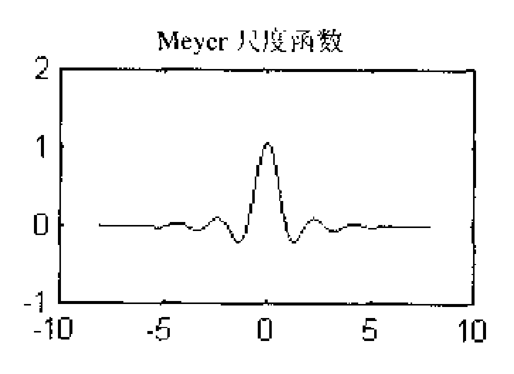


图　2.12

参见：meyeraux，wavefun，waveinfo

7. meyeraux

功能：Meyer 小波辅助函数

格式：meyeraux(X)

说明：该函数返回作为 meyer 函数用的辅助函数在向量或矩阵 X 各点的值。关于辅助函数的内容，参见第 1.2 节，函数为

$$35x^4 - 84x^5 + 70x^6 - 20x^7$$

举例：

```
x=[1,2,3;4,5,6;7,8,9]
y=meyeraux(x)
```

输出结果：

```
x =
     1     2     3
     4     5     6
     7     8     9
y =
            1          -208        -10287
      -118016       -709375      -2940624
     -9563183     -26202112     -63188991
```

参见：meyer

8. morlet

功能：Morlet 小波

格式：[PSI,X]=morlet(LB,UB,N)

说明：该函数返回一个有效支撑为[LB,UB]，在有效支撑上有 N 个均匀分布点的 Morlet 小波。输出参数为在 X 上的 psi(即 Ψ)函数的值。该函数具有的有效支撑为[−4，4]。

$$\Psi(x) = \mathrm{e}^{-x^2/2}\cos 5x$$

举例：lb=−4；ub=4；n=1000；

```
%计算并画出 Morlet 小波函数
[psi,x]=morlet(lb,ub,n);
subplot(221); plot(x,psi); title('Morlet 小波');
```

输出结果(如图 2.13 所示)。

参见：waveinfo





图　2.13

9. symwavf

功能：Symlets 小波滤波器

格式：F=symwavf('symname')

说明：该函数返回与所指定的 symlet 小波相关联的尺度滤波器，N 的可能值为 2，3，4，5，6，7 或 8。

举例：%计算与 sym4 相关的尺度滤波器

```
w=symwavf('sym4')
```

输出结果：

```
w =
  Columns 1 through 7
    0.0228   -0.0089   -0.0702    0.2106    0.5683    0.3519   -0.0210
  Column 8
   -0.0536
```

参见：waveinfo

2.3　一维小波变换

在本节里，我们将详细讲述有关一维小波变换的有关函数的用法。一维小波变换包括一维小波连续变换和一维离散小波变换。

1. cwt

功能：一维连续小波变换

格式：① coefs=cwt(s,scales,'wname')

② coefs=cwt(s,scales,'wname','plot')

说明：该函数为一维连续小波分析函数，其中 coefs 为连续小波变换后的返回系数 $W_f(a,b)$

矩阵，系数以行方向存储在矩阵中。$W_f(a,b)$定义为

$$W_f(a,b) = \langle f, \Psi_{a,b} \rangle = |a|^{-1/2}\int_R f(t)\overline{\Psi\left(\frac{t-b}{a}\right)}\mathrm{d}t$$

因为 $f(t)$是一个离散的信号，所以我们可以将信号用 $f(k)$的形式表示，$k=1$ 到 length(f)，所以对于一具体的尺度 a，cwt 将计算 b 从 1 到 length(f)所对应的每一个系数值$W_f(a,b)$。

s 为待分析的信号，在计算中通常是以离散的形式给出，如在 MATLAB 自身所带有的 noissin 信号等。scales 为连续小波变换的尺度向量，wname 为小波函数名。如果尺度为离散的值 a1,a2,a3…,则尺度向量 scales 可以表示为：[a1,a2,a3 …]的形式，值与值之间用逗号或者空格隔开；倘若尺度向量由起始尺度 amin、终止尺度 amax、尺度步长 astep 三者组成，则尺度向量 scales 可以表示为：[amin:astep:amax]的形式(中括号“[ ]”此时可省略)；倘若尺度向量同时包括上面两种情况，则把离散尺度写在前面，步长表示方式写在后面，如：[a1,a2,a3,amin:astep:amax]。wname 是小波函数名，如：haar,db1,db2,meyer 等。Coefs 是指返回系数矩阵(小波变换系数)，矩阵的行数为小波变换中尺度的个数，列数为信号采样点的个数，如 noissin 信号的采样点个数为 1000，矩阵第一行的值对应第一个尺度变换后的系数，第二行的值对应第二个尺度变换后的系数，以后依此类推。对于格式②，“plot”是用来画出小波变换后系数的图形，在图形中，系数的大小是以灰度的深浅来表示，颜色越深，则变换后的系数 $W_f(a,b)$越大。

另外，(1)尺度必须为正实数。(2)逗号可以换成空格的形式。(3)当尺度步长 astep 没有显式给出时，则表示 astep 取默认步长 1。

连续小波变换 cwt 的算法说明：

由连续小波变换的定义

$$W_f(a,b) = \langle f, \Psi_{a,b} \rangle = |a|^{-1/2}\int_R f(t)\overline{\Psi\left(\frac{t-b}{a}\right)}\mathrm{d}t$$

有

$$W_f(a,b) = \sum_k \int_k^{k+1} f(t)|a|^{-1/2}\overline{\Psi\left(\frac{t-b}{a}\right)}\mathrm{d}t$$

因为 $s(t)=s(k)(t\in[k,k+1])$，所以，

$$\begin{aligned} W_f(a,b) &= |a|^{-1/2}\sum_k \int_k^{k+1} f(k)\overline{\Psi\left(\frac{t-b}{a}\right)}\mathrm{d}t \\ &= |a|^{-1/2}\sum_k f(k)\left(\int_{-\infty}^{k+1}\overline{\Psi(\frac{t-b}{a})}\mathrm{d}t - \int_{-\infty}^{k}\overline{\Psi(\frac{t-b}{a})}\mathrm{d}t\right) \end{aligned}$$

所以，对任何一个尺度 a，可以通过计算信号 $f(k)$与小波函数 $\Psi(t)$之间的卷积来计算小波系数 $W_f(a,b)$，其中，$b=1, 2, \cdots,$ length(f)。

举例：对 MATLAB 中所带有的 noissin 信号进行连续小波变换，尺度 a 分别为 0.12,0.24,0.48,1.2,2,4,6,8,10，小波函数用 db3,请求出连续小波变换后的系数。

程序：

```
load noissin; %装载信号
s=noissin(1 : 100);
ls=length(s); %计算信号点的个数 ls
```

```
%对s进行一维连续小波变换，把返回系数存到矩阵w中
w=cwt(s,[12.12,10.24,15.48,1.2,2:2:10],'db3','plot')
Xlabel('时间');
Ylabel('变换尺度');
title('对应于尺度a=0.12,0.24…小波变换系数的绝对值');
```

输出结果：

执行程序后，返回矩阵为一个9×1000矩阵。在此为节省篇幅，我们不将结果打印出来，读者可自己上机运行该程序观察结果。变换后，不同尺度对应的灰度值图如图2.14所示。

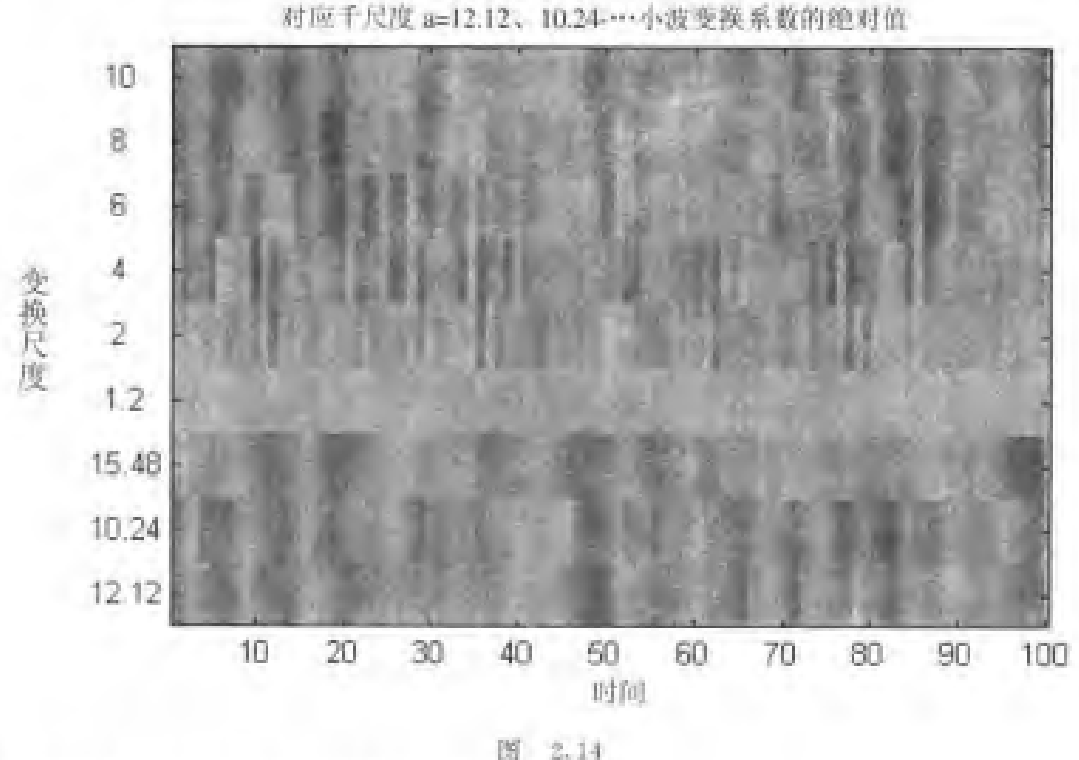


图 2.14

参见：wavedec，wavefun，waveinfo，wcodemat

2. dwt

功能：单尺度一维离散小波变换

格式：① [cA,cD]=dwt(X,'wname')

② [cA,cD]=dwt(X,Lo_D,Hi_D)

说明：该函数用于进行一维离散小波分解，X为被分析的离散信号，wname为分解所用到的小波函数，Lo_D，Hi_D为分解滤波器，cA和cD分别为返回的低频系数和高频系数向量，它们长度相等且为：length(X)/2（当length(X)为偶数时）或(length(X)+1)/2（当length(X)为奇数时）。

如果令lx=length(X)，lf=length(Lo_R)，则有

$$\text{length}(cA) = \text{length}(cD) = \text{floor}(lx + lf - 2)/2)$$

举例：load noissin；%装载原始一维信号

```
s=noissin(1 : 1000);
%画出原始信号的波形
subplot(411); plot(s);
title('原始信号');
%下面用 haar 小波函数进行一维离散小波变换
[ca1,cd1]=dwt(s,'haar');
subplot(4,2,3); plot(ca1);
Ylabel('haar(ca1)');
subplot(4,2,4); plot(cd1);
Ylabel('haar(cd1)');
%给定一个小波 db2，计算与之相关的分解滤波器
[Lo_D,Hi_D]=wfilters('db2','d');
%用分解滤波器 Lo_D,Hi_D 计算信号 s 的离散小波分解系数
[ca2,cd2]=dwt(s, Lo_D,Hi_D);
subplot(4,2,5); plot(ca2);
Ylabel('db2(ca2)');
subplot(4,2,6); plot(cd2);
Ylabel('db2(cd2)');
```

输出结果(如图 2.15 所示)：

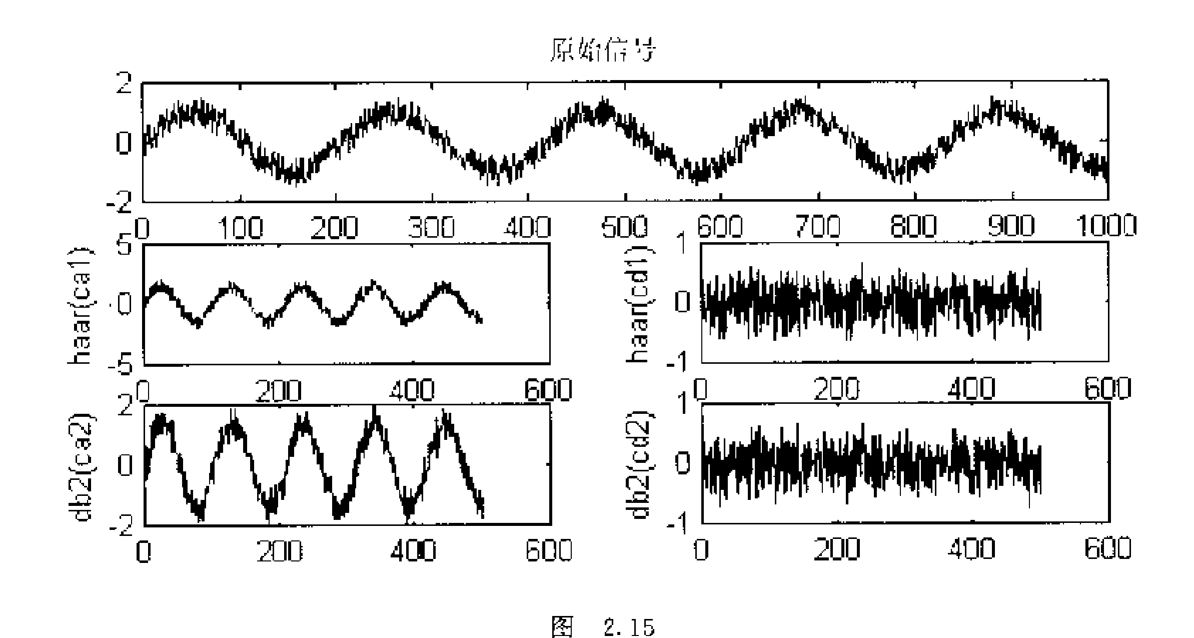


图　2.15

参见：dwtmode，dwtper，idwt，wavedec，waveinfo

3. dwtmode

功能：离散小波变换拓展模式

格式：① dwtmode

② dwtmode('mode')

说明：当对信号或图像的边缘进行处理时，需要信号的边缘进行拓展，一般说来，拓展模式有三种。dwtmode 函数就是在对信号(或图像)进行离散小波变换或小波包变换时，进行模式拓展设置，不同的模式代表了对待信号(或图像)边缘不同的处理方法。各种模式的具体意义见表 2-2。

表 2-2　dwtmode 函数的模式类型

模式类型	类型说明
Zdp	补零模式，这是一种缺省的类型模式
Sym	对称延拓模式，即把边缘值进行复制
Spd	平滑模式，即对边缘进行某种平滑处理

举例：
```
load leleccum;
s=leleccum(1 : 1000); w='db3';
lx=length(s);
subplot(621); plot(s); %画出原始信号的波形图
title('原始信号');
dwtmode;
[cazpd,cdzpd]=dwt(s,w);
lxtzpd—2 * length(cazpd)
%下面进行信号的重构
xzpd=idwt(cazpd,cdzpd,w,lx);
subplot(622); plot(xzpd); %画出重构后的波形图
title('zpd 模式重构图');
dwtmode('sym');
[casym,cdsym]=dwt(s,w);
lxtsym=2 * length(casym)
%下面进行信号的重构
xsym=idwt(casym,cdsym,w,lx);
subplot(625); plot(xsym); %画出重构后的波形图
title('sym 模式重构图');
dwtmode('spd');
[caspd,cdspd]=dwt(s,w);
lxtspd=2 * length(caspd)
%下面进行信号的重构
xspd=idwt(caspd,cdspd,w,lx);
subplot(626); plot(xspd); %画出重构后的波形图
title('spd 模式重构图');
```
输出结果(如图 2.16 所示)：

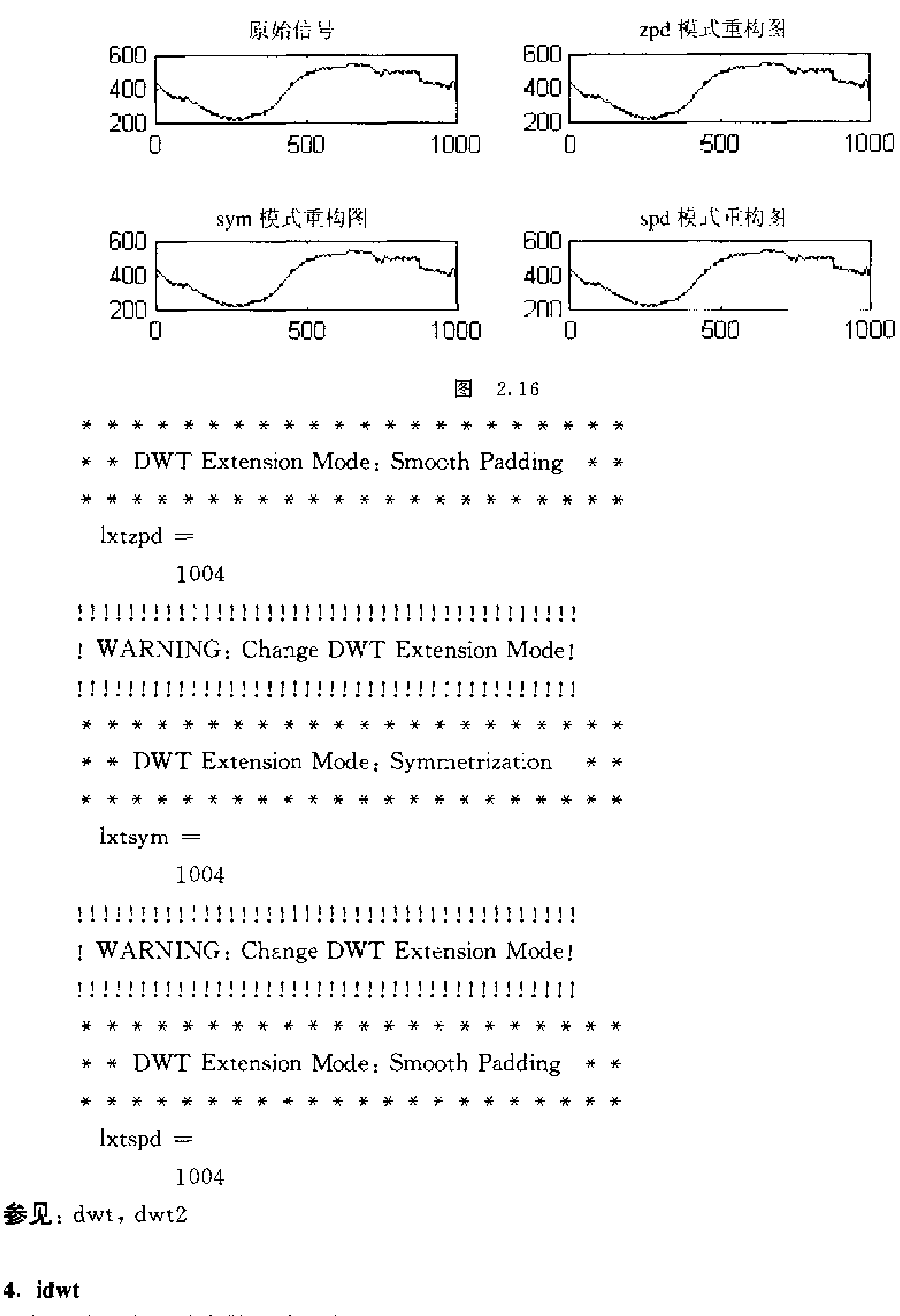


图　2.16

```
* * * * * * * * * * * * * * * * * * * * * *
* * DWT Extension Mode: Smooth Padding   * *
* * * * * * * * * * * * * * * * * * * * * *
 lxtzpd =
        1004
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! WARNING: Change DWT Extension Mode!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
* * * * * * * * * * * * * * * * * * * * * *
* * DWT Extension Mode: Symmetrization    * *
* * * * * * * * * * * * * * * * * * * * * *
 lxtsym =
        1004
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! WARNING: Change DWT Extension Mode!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
* * * * * * * * * * * * * * * * * * * * * *
* * DWT Extension Mode: Smooth Padding   * *
* * * * * * * * * * * * * * * * * * * * * *
 lxtspd =
        1004
```

参见：dwt，dwt2

4. idwt

功能：单尺度一维离散小波逆变换

格式：① X=idwt(cA,cD,'wname')

② X=idwt(cA,cD,Lo_R,Hi_R)

③ X=idwt(cA,cD,'wname',L)

④ X=idwt(cA,cD,Lo_R,Hi_R,L)

说明：该函数用于单尺度一维离散小波变换的重构。对格式①、③，它是用小波函数进行重构，对于②、④它是用重构滤波器进行重构，其中，cA 和 cD 的长度是相等的，Lo_R 和 Hi_R 的长度是相等的。返回系数 X 为重构后信号的向量。如果 cA 的长度为 la，Lo_R 的长度为 lf，则 X 的长度为 length(X)=$2*la-lf+2$。对于格式③、④，则是对信号中间长度为 L 的部分进行重构，L<$2*la-lf+2$。

运用重构滤波器进行重构的步骤可以用图 2.17 方框图描述。

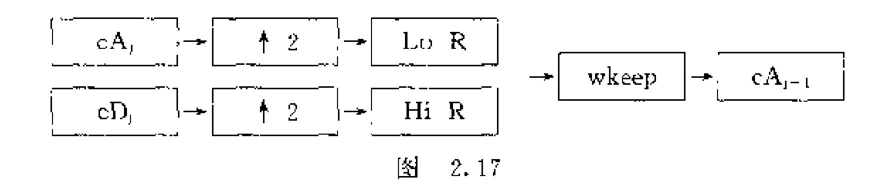


图 2.17

其中，

[1 ↑ 2] 表示向向量奇数位置插入 0 值；

[X] 表示与滤波器 X 进行卷积运算，X 可以为 Lo_R 或 Hi_R；

[wkeep] 表示提取向量中间适当的长度。

举例：

```
load leleccum; %装载原始一维信号
s=leleccum(1:1000);
subplot(6,2,1); plot(s);
title('原始信号');
%下面用 db2 小波函数进行一维离散小波变换
[ca,cd]=dwt(s,'db2');
%用小波函数 db2 进行信号的重构
x1=idwt(ca,cd,'db2');
subplot(625); plot(x1);
title('小波重构');
errx1max=max(abs(s-x1))
errx1=s-x1;
subplot(626); plot(errx1);
title('小波重构误差');
axis([0,1000,-2e-11,2e-11]);
%产生一个与 db2 相关的重构滤波器
[Lo_D,Hi_D,Lo_R,Hi_R]=wfilters('db2');
%用分解滤波器对信号 s 进行一维小波分解
[ca,cd]=dwt(s,Lo_D,Hi_D);
%用重构滤波器进行信号的重构
x2=idwt(ca,cd,Lo_R,Hi_R);
```

```
subplot(6,2,9); plot(x2);
title('滤波器重构');
errx2max=max(abs(s-x2))
errx2=s-x2;
subplot(6,2,10); plot(errx2);
title('滤波器重构误差');
axis([0,1000,-2e-11,2e-11]);
```



输出结果(如图 2.18 所示):

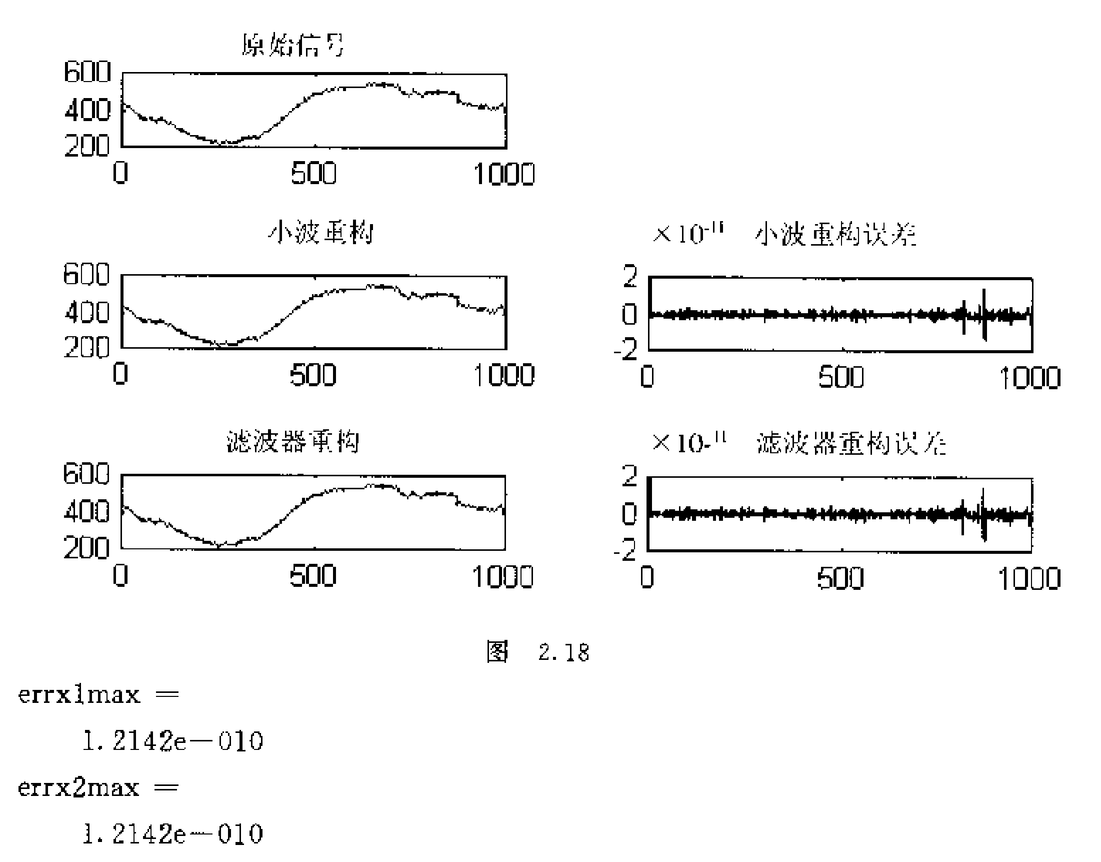


图　2.18

```
errx1max =
    1.2142e-010
errx2max =
    1.2142e-010
```

从上面的分析可以看出，采用小波重构和与小波重构滤波器进行重构，其效果基本上是一样的，它们在重构精度上存在着较小的差别。

参见：dwt，idwtper，upwlev

5. dwtper

功能：单尺度一维离散小波变换(周期性)

格式：① [cA,cD]=dwtper(X, 'wname')

② [cA,cD]=dwtper(X, Lo_D, Hi_D)

说明：该函数是一个一维小波分析函数，它用来计算信号 X 的周期小波分解的低频系数向量 cA 和高频系数向量 cD。在分解时可以采用小波函数进行分解(如格式①)，也可以采用分解滤波器进行 分解(如格式②)。cA 与 cD 向量的长度为：

length(X)为奇数时：length(cA)=length(cD)=(length(X)+1)/2

length(X)为偶数时：length(cA)=length(cD)=length(X)/2

算法说明：周期性小波变换(dwtper)的算法同 dwt 一样，只不过信号 X 需要经过周期性的拓展。更确切一点说，当 lx=length(X)为偶数时，信号拓展成为

$$\text{extX} = [X(lx - lf + 1), X, X(1 : lf)]$$

其中，lf 为滤波器的长度。信号拓展后，用一般的卷积和抽样进行运算，最后只保留中间长度为 $lx/2$ 的部分即可。

举例：
```
load leleccum; %装载原始一维信号
s=leleccum(1 : 2000);
%画出原始信号的波形
subplot(611); plot(s);
title('原始信号');
%下面用 haar 小波函数进行一维离散小波变换
[ca1,cd1]=dwtper(s,'haar');
subplot(6,2,5); plot(ca1);
title('用 haar 小波分解的低频系数 ca1');
subplot(6,2,6); plot(cd1);
title('用 haar 小波分解的高频系数 cd1');
axis([0,1000,-20,20]);
%给定一个小波 dh2，计算与之相关的分解滤波器
[Lo_D,Hi_D]=wfilters('db2','d');
%用分解滤波器 Lo_D,Hi_D 计算信号 s 的周期性离散小波分解系数
[ca2,cd2]=dwtper(s, Lo_D,Hi_D);
suhplot(6,2,9); plot(ca2);
title('用分解滤波器分解的低频系数 ca2');
subplot(6,2,10); plot(cd2);
title('用分解滤波器分解的高频系数 ca2');
axis([0,1000,-20,20]);
```

输出结果(如图 2.19 所示)：

参见：dwt, idwtper

6. idwtper

功能：单尺度一维离散小波重构(周期性)

格式：① X=idwtper(cA,cD,'wname')

② X=idwtper(cA,cD,Lo_R,Hi_R)

③ X=idwtper(cA,cD,'wname',L)

④ X=idwtper(cA,cD,Lo_R,Hi_R,L)

说明：该函数用于单尺度一维离散小波分解(周期性)的重构。对格式①、③，它是用小波函数进行重构，对于②、④它是用重构滤波器进行重构，其中，cA 和 cD 的长度是相等的，Lo_R 和 Hi_R 的长度是相等的。返回系数 X 为重构后信号的向量。如果 cA 的长度为 la，

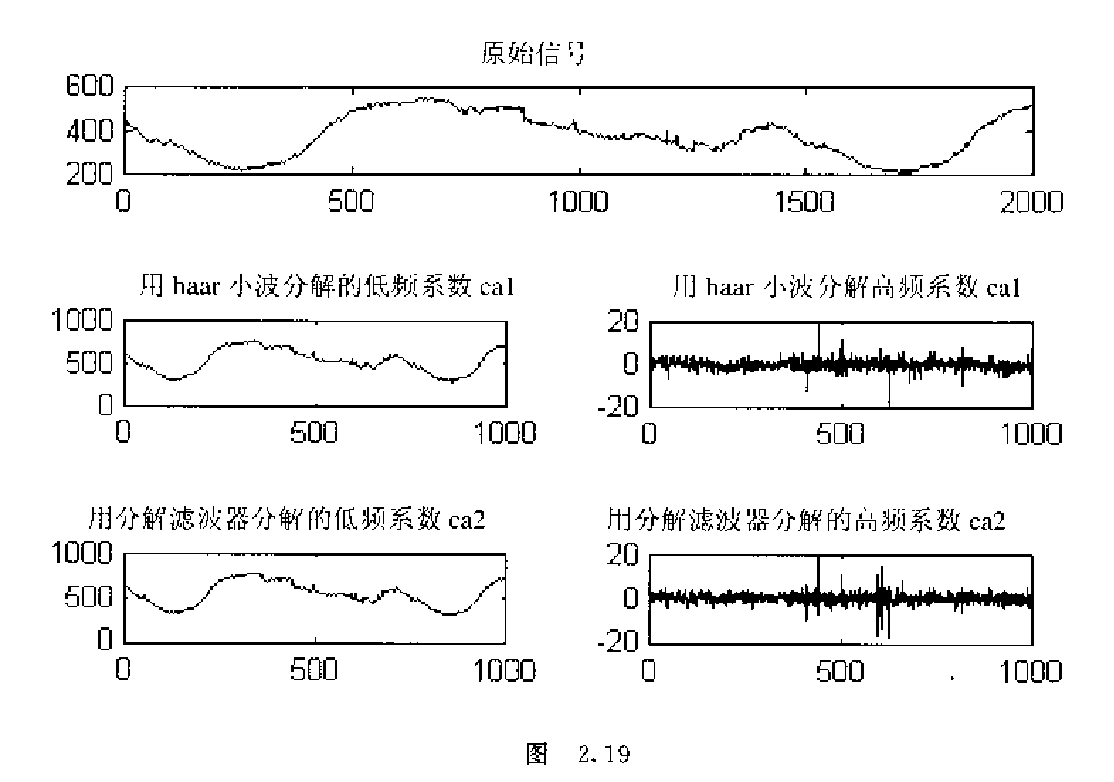


图　2.19

则 X 的长度为 length(X)=2 * *la*。对于格式③、④，则是对信号中间长度为 L 的部分进行重构，L≤length(X)。

举例：%初始化信号并获得一组重构滤波器

```
x=sin(0.3*[1:300]); lx=length(x); w='db9';
subplot(6,2,1); plot(x); %画出原始信号的波形图
title('原始信号');
%产生与 db9 小波相应的分解与重构滤波器
[Lo_D,Hi_D,Lo_R,Hi_R]=wfilters(w);
%下面用 dwt 函数对信号进行离散小波分析
[ca1,cd1]=dwt(x,w);
%下面进行信号的重构
x1=idwt(ca1,cd1,w,lx);
subplot(6,2,5); plot(x1); %画出重构后的波形图
title('idwt 重构信号');
%求出重构误差、最大的误差并画出波形
errx1=x-x1;
errx1max=max(abs(x-x1))
subplot(6,2,6); plot(errx1);
title('idwt 重构误差');
%下面运用周期性离散小波分析与上面进行对比分析
[ca2,cd2]=dwtper(x,w);
%进行信号的重构
```

```
x2=idwtper(ca2,cd2,w,lx);
subplot(6,2,9); plot(x2); %画出重构后的波形图
title('idwtper 重构信号');
%求出重构误差、最大的误差并画出波形
errx2=x-x2;
errx2max=max(abs(x-x2))
subplot(6,2,10); plot(errx2);
title('idwtper 重构误差');
```

输出结果(如图 2.20 所示):

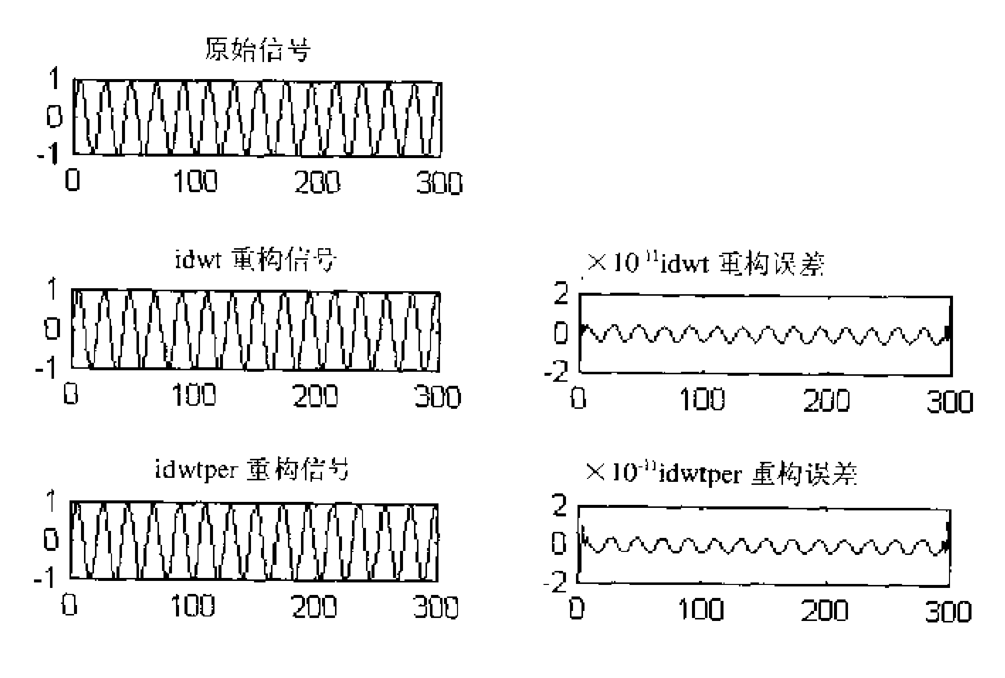


图 2.20

```
errx1max =
    1.5577e-011
errx2max =
    1.4600e-011
```

参见: dwtper

7. wavedec

功能: 多尺度一维小波分解(一维多分辨分析函数)

格式: ① [C,L]=wavedec(X, N, 'wname')

② [C,L]=wavedec(X, N, Lo_D, Hi_D)

说明: wavedec 函数用小波或分解滤波器完成对信号 X 的一维多尺度分解，N 为尺度，且是严格的正整数。输出参数 C 是由[cAj, cDj, cD j-1, …, cD1]组成，L 是由[cAj 的长度，cDj 的长度，cD j-1的长度，…，cD1 的长度，X 的长度]组成。以一个 3 尺度分解为例，其分解结构的组织形式如图 2.21 所示。

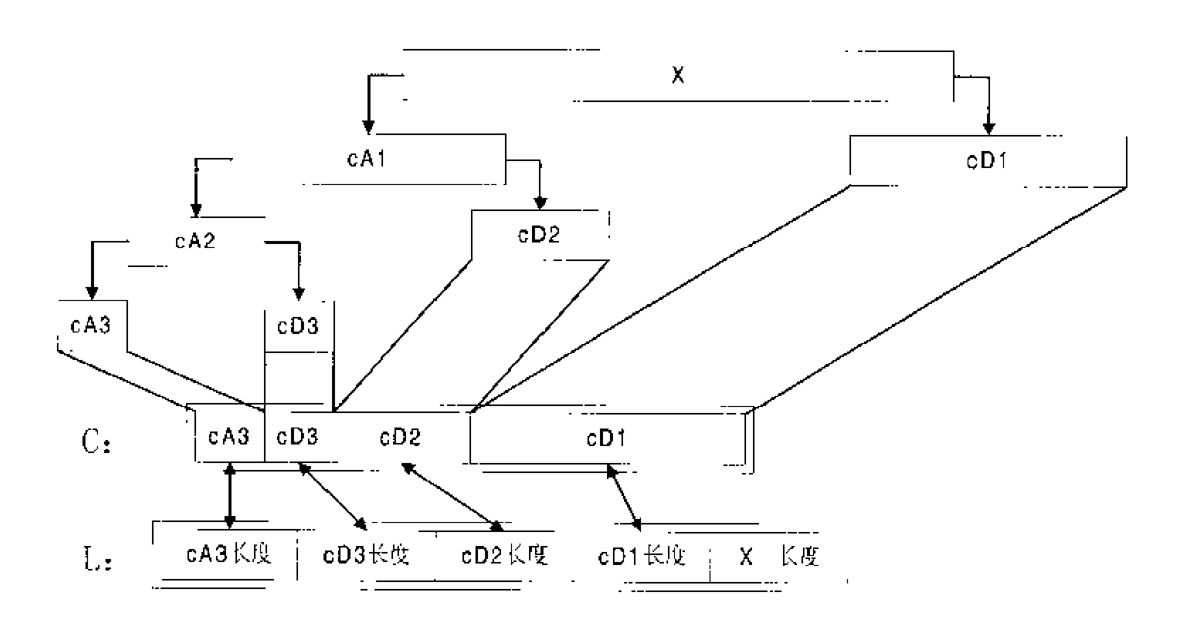


图 2.21

格式②返回如上图所示的分解结构。

说明：给定一个长度为 N 的信号 S，离散小波分解(DWT)最多可以把信号分解成 $\log_2^N$ 个频率级。第一步分解开始于信号 S，分解后分解系数由两部分组成：低频系向量 cA1 和高频系数向量 cD1，向量 cA1 是由信号 S 与低通分解滤波器 Lo_D 经过卷积运算得到，向量 cD1 是由信号 S 与高通分解滤波器 Hi_D 经过卷积运算得到。为了更加清楚说明第一步分解，其分解过程如图 2.22 所示。

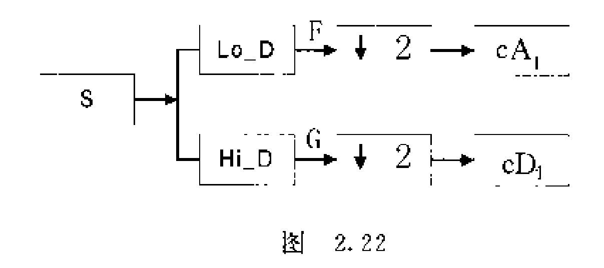


图 2.22

其中，

X 表示信号与滤波器的卷积；

↓ 2 表示抽取向量中标号为偶数的元素。

每个滤波器的长度均为 2N，如果 n=length(S)，则信号 F 和 G 的长度为：n+2N−1，所以系数 cA1 和 cD1 的长度为 floor$(\frac{n-1}{2})$+N。

在下一步分解，用同样的方法把低频系数 cA1 分成两部分，即把上面方框图中的 S 用 cA1 代替，分解后返回尺度 2 的低频系数 cA2 和高频系数 cD2。再往以下分解依此类推。

举例：load sumsin；s=sumsin；

```
subplot(611)；plot(s)；%画出原始信号的波形
title('原始信号')；
```

```
%用小波函数 db1 对信号进行 3 尺度的小波分解
[c,l]=wavedec(s,3,'db1');
subplot(613); plot(c); %画出变换后的波形
title('信号 S 的 3 尺度小波分解结构');
Xlabel('尺度 3 的低频系数和尺度 3、2 和 1 的高频系数');
axis([0,1000,-5,5]);
```

输出结果(如图 2.23 所示)：

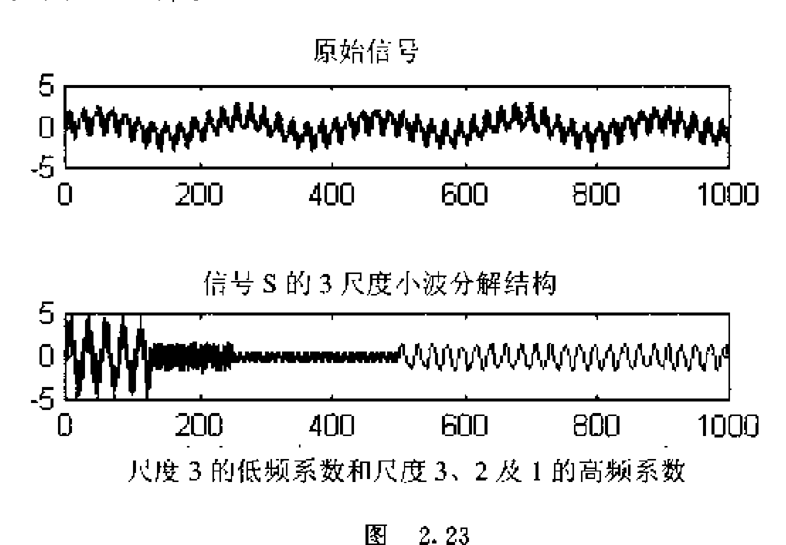


图 2.23

参见：dwt，waveinfo，wfilters，wmaxlev

8. appcoef

功能：提取一维小波变换低频系数

格式：① A=appcoef(C, L, 'wname', N)

② A=appcoef(C, L, 'wname')

③ A=appcoef(C, L, Lo_R, Hi_R)

④ A=appcoef(C, L, Lo_R, Hi_R, N)

说明：该函数是一个一维小波分析函数，它用于从小波分解结构[C,L]中提取一维信号的低频系数，其中，[C,L]为小波分解结构，wname 为小波函数，N 为尺度。格式①计算尺度 N(N 必须为一个正整数且 0≤N≤length(L)-2)时的一维分解低频系数；格式②用于提取最后一尺度(尺度 N=length(L)-2)的小波变换低频系数；格式③、④是用滤波器 Lo_R 和 Hi_R 进行信号低频系数的提取，一般说来，该滤波器是与某一小波函数相关的，通常可由 wfilters 函数得到。返回系数 A 是一个向量，向量的元素个数为 lengtb(S)/2^N。

举例：%下面装载一维信号

```
load leleccum; s=leleccum(1:2000); ls=length(s);
subplot(421); plot(s); title('原始信号');
%尺度为 3，小波函数为 db1 时进行分解
[c,l]=wavedec(s,3,'db1');
%从小波分解结构[c,l]中提取尺度 1、2 的低频系数
```

```
ca1=appcoef(c,l,'db1',1);
subplot(4,4,5); plot(ca1);
Ylabel('ca1');
ca2=appcoef(c,l,'db1',2);
subplot(4,8,17);plot(ca2);
Ylabel('ca2');
```

输出结果(如图 2.24 所示):

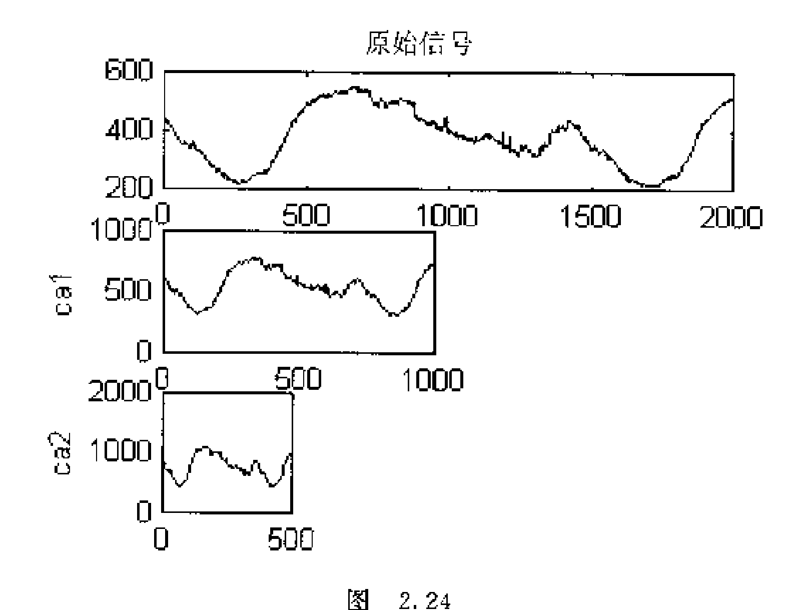


图 2.24

参见: detcoef, wavedec

9. detcoef

功能: 提取一维小波变换高频系数

格式: ① D=detcoef(C, L, N)

② D=detcoef(C, L,)

说明: 该函数是一个一维小波分析函数,它与 appcoef 函数相对应,用来计算一维小波变换后的高频系数。格式①提取尺度为 N(N 必须为一个正整数且 0≤N≤length(L)-2),分解结构为[C, L]的一维分解高频系数;格式②用于提取最后一尺度(尺度 N=length(L)-2)的一维分解高频系数。该函数返回一个向量,且其长度为 length(S)/2^N。

举例: %下面装载一维信号

```
load leleccum; s=leleccum(1 : 2000); ls=lengtb(s);
%画出原始信号波形
subplot(421); plot(s);
title('原始信号');
%尺度为 3,小波函数为 db1 时进行分解
[c,l]=wavedec(s,3,'db1');
%从小波分解结构[c,l]中提取尺度分别为 1, 2 的高频系数
```

```
cd1=detcoef(c,l,1);
subplot(4,4,5); plot(cd1);
Ylabel('cd1');
cd2=detcoef(c,l,2);
subplot(4,8,17); plot(cd2);
Ylabel('cd2');
```

输出结果(如图 2.25 所示):

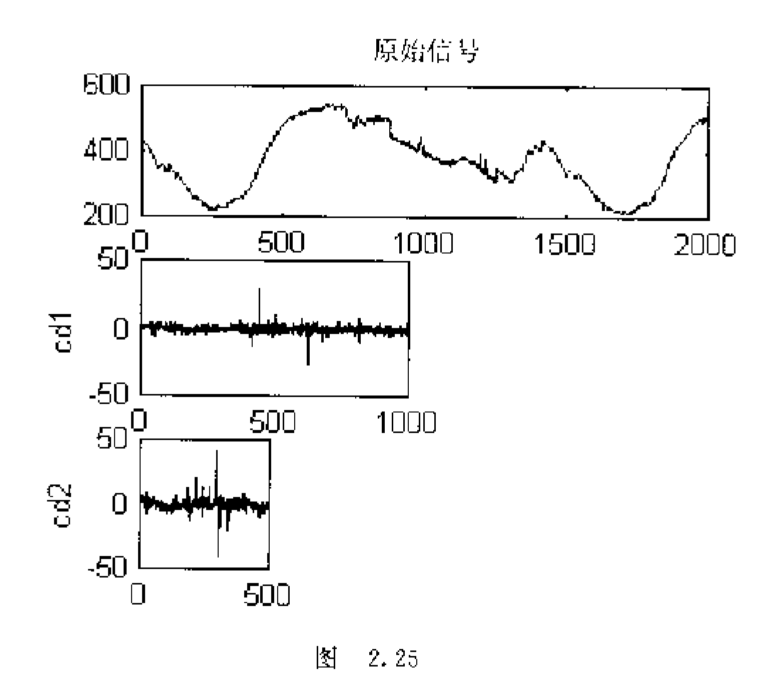


图　2.25

参见: appcoef, wavedec

10. waverec

功能: 多尺度一维小波重构

格式: ① X=waverec(C,L,'wname')

② X=waverec(C,L,Lo_R,Hi_R)

说明: 该函数是用指定的小波函数或重构滤波器对小波分解结构[C, L]进行多尺度一维小波重构，它是 wavedec 函数的逆函数，即有

X=waverec(wavedec(X,N,'wname'),'wname')

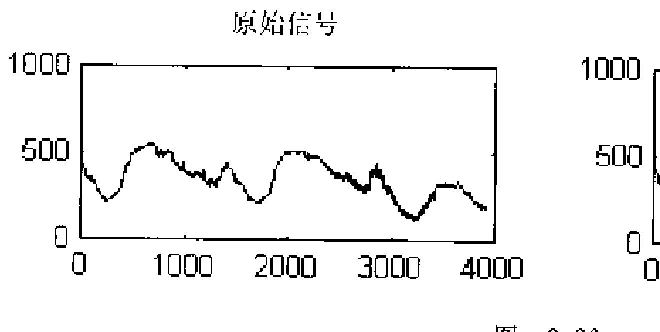
另外，X=waverec(C,L,'wname')与 X=appcoef(C,L,'wname',0)具有等价性。格式①是用小波函数进行重构，格式②是用重构滤波器进行重构。

举例: %装载一个一维信号

```
load leleccum; s=leleccum(1:3920); ls=length(s);
subplot(211); plot(s); %画出原始信号的波形图
title('原始信号');
%用小波函数 db5 对信号进行 3 尺度分解
[c,l]=wavedec(s,3,'db5');
```

```
%在小波分解结构[c,l]的基础上对信号进行重构
a=waverec(c,l,'db5');
subplot(212); plot(a); %画出重构后的信号波形图
title('重构信号');
%检测重构的误差
err=norm(s-a)
```

输出结果(如图 2.26 所示):



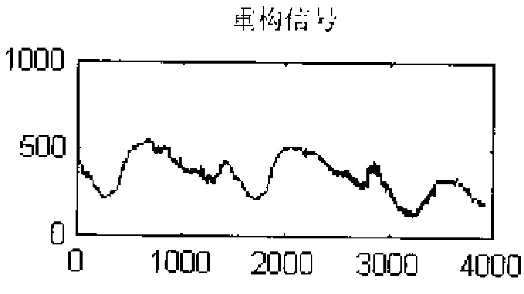


图 2.26

```
err =
  3.2632e-009
```

参见: appcoef, idwt, wavedec

11. upwlev

功能: 单尺度一维小波分解的重构

格式: ① [NC,NL,cA]=upwlev(C,L,'wname')

② [NC,NL,cA]=upwlev(C,L,Lo_R,Hi_R)

说明: 该函数用于对小波分解结构[C,L]进行单尺度重构，返回上一尺度的分解结构[NC,NL]并提取最后一尺度的低频系数向量 cA，即如果[C,L]是尺度 N 的分解结构，则[NC,NL]是尺度 N−1 的分解结构。wname 指定所用到的小波，[C,L]是原始的分解结构。格式①是用小波进行重构，格式②是用重构滤波器进行重构。

举例: %装载一个一维信号

```
load sumsin; s=sumsin;
%用小波函数对信号进行第三层分解(尺度 3)
[c,l]=wavedec(s,3,'db1');
subplot(611); plot(s);
title('原始信号');
subplot(613); plot(c);
title('尺度 3 的小波分解结构');
Xlabel('尺度 3 的低频系数和尺度 3、2、1 的高频系数');
%下面用 upwlev 函数重构，得到尺度 2 的小波分解结构
[nc,nl]=upwlev(c,l,'db1');
subplot(615); plot(nc);
```

```
title('尺度 2 的小波分解结构');
Xlabel('尺度 2 的低频系数和尺度 2、1 的高频系数');
```

输出结果(如图 2.27 所示)：

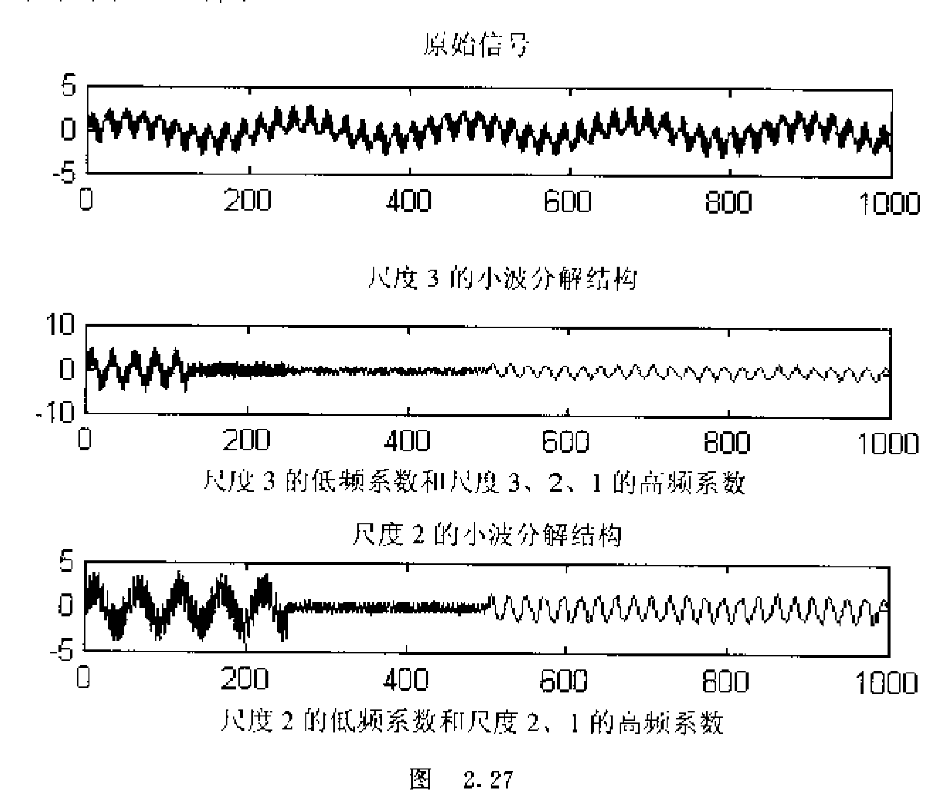


图　2.27

参见：idwt，upcoef，wavedec

12. wrcoef

功能：对一维小波系数进行单支重构

格式：① X=wrcoef('type',C,L,'wname',N)

② X=wrcoef('type',C,L,Lo_R,Hi_R,N)

③ X=wrcoef('type',C,L,'wname')

④ X=wrcoef('type',C,L,Lo_R,Hi_R)

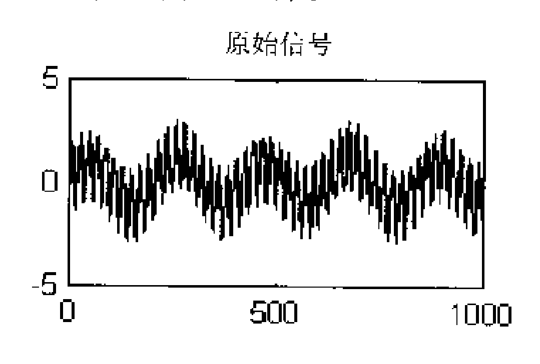
说明：该函数是对一维信号的分解结构[C，L]用指定的小波函数或重构滤波器进行重构。当 type=a 时，指对信号的低频部分进行重构，此时 N 可以为 0；当 type=d 时，指对信号的高频部分进行重构，此时 N 为严格的正整数。

举例：%装载一个一维信号

```
load sumsin; s=sumsin;
subplot(321); plot(s); %画出原始信号的波形图
title('原始信号');
%用小波函数 sym4 对信号进行 5 尺度小波分解
[c,l]=wavedec(s,5,'sym4');
%对尺度 5 上的低频部分进行重构
a5=wrcoef('a',c,l,'sym4',5);
```

```
subplot(322); plot(a5); %画出重构后的波形图
title('重构信号');
```

输出结果(如图 2.28 所示)：



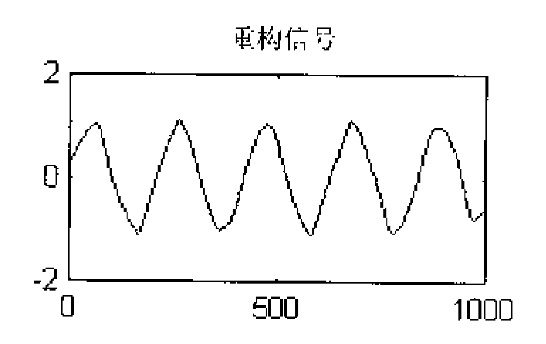


图 2.28

参见：appcoef，detcoef，wavedec

13. upcoef

功能：一维系数的直接小波重构

格式：① Y=upcoef(0,X,'wname',N)

② Y=upcoef(0,X,'wname',N,L)

③ Y=upcoef(0,X,Lo_R,Hi_R,N)

④ Y=upcoef(0,X,Lo_R,Hi_R,N,L)

⑤ Y=upcoef(0,X,'wname')

⑥ Y=upcoef(0,X,Lo_R,Hi_R)

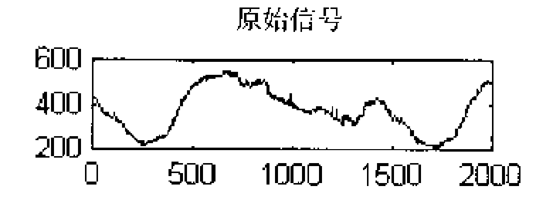
说明：该函数是用于一维小波分析的函数，它用于计算向量 X 向上 N 步的重构小波系数，N 是严格的正整数。如果 0=a,则是对低频系数进行重构，如果 0=d，则是对高频系数进行重构。对于格式②、④的情况，则是对向量 X 中间长度为 L 的部分进行重构。在重构的过程中，格式①、②、⑤是用小波函数进行重构，格式③、④、⑥是用重构滤波器进行重构。

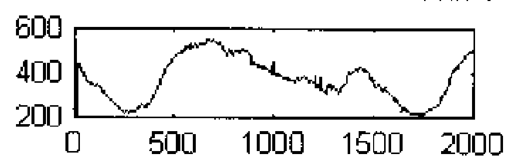
另外需要注意的是，格式⑤等价于 Y=upcoef(0,X,'wname',1)；格式⑥等价于 Y=upcoef(0,X,Lo_R,Hi_R,1)。

举例：
```
load leleccum;
s=leleccum(1:2000);
subplot(6,2,1); plot(s);
title('原始信号');
[c,l]=wavedec(s,3,'db6');
ca1=appcoef(c,l,'db6',1);
sca1=upcoef('a',ca1,'db6',1);
subplot(6,2,2); plot(sca1);
title('尺度 1 低频系数 ca1 向上 1 步重构信号');
axis([0,2000,200,600]);
```

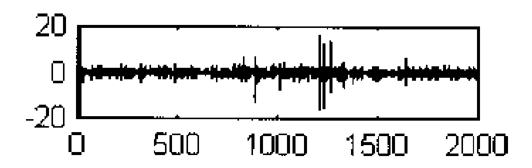
```
sca1L=upcoef('a',ca1,'db6',1,1000);
subplot(6,2,5); plot(sca1L);
title('ca1 向上 1 步重构并只取中间 1000 个点');
axis([0,2000,200,600]);
cd1=detcoef(c,l,1);
scd1=upcoef('d',cd1,'db6',1);
subplot(6,2,6); plot(scd1);
title('尺度 1 高频系数 cd1 向上 1 步重构信号');
axis([0,2000,-20,20]);
%产生与 db6 小波相应的滤波器
[Lo_R,Hi_R]=wfilters('db6','r');
ca2=appcoef(c,l,'db6',2);
sca2=upcoef('a',ca2,Lo_R,Hi_R,2);
subplot(6,2,9); plot(sca2);
title('尺度 2 低频系数 ca2 向上 2 步重构信号');
axis([0,2000,200,600]);
cd2=detcoef(c,l,2);
scd2=upcoef('d',cd2,'db6',2);
subplot(6,2,10); plot(scd2);
title('尺度 2 高频系数 cd2 向上 2 步重构信号');
axis([0,2000,-20,20]);
```

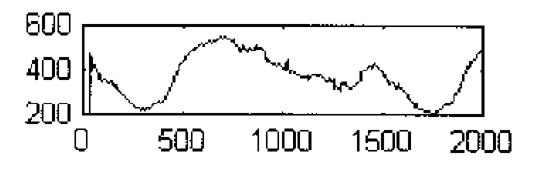
输出结果(如图 2.29 所示):











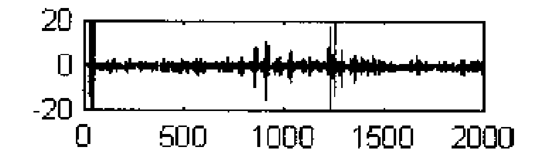


图　2.29

参见：idwt

2.4　二维小波变换

本节我们将详细讲述二维小波变换的有关函数的用法。

1. dwt2

功能：单尺度二维离散小波变换

格式：① [cA,cH,cV,cD]=dwt2(X, 'wname')

　　　② [cA,cH,cV,cD]=dwt2(X, Lo_D, Hi_D)

说明：该函数用于进行二维离散小波分解，X 为被分析的离散信号，wname 为分解所用到的小波函数，Lo_D、Hi_D 为分解滤波器，cA 和 cH、cV、cD(水平、垂直、对角线)分别为返回的低频系数和高频系数向量。Lo_R 和 Hi_R 的长度必须相等。如果 sx=size(X)、lf=size(Lo_R)(或 Hi_R)，则 size(cA)=size(cH)=size(cV)=size(cD)=floor(($sx+lf-1$)/2)。

对于图像分解，其算法同一维分解的情况类似，二维小波函数和尺度函数是通过一维小波函数和尺度函数经过张量积(tensor product)变换得到。二维小波分解是把尺度 j 的低频部分分解成四个部分：尺度 j+1 的低频部分和三个方向(水平、垂直、斜线)的高频部分。其基本的分解步骤可用图 2.30 表示。

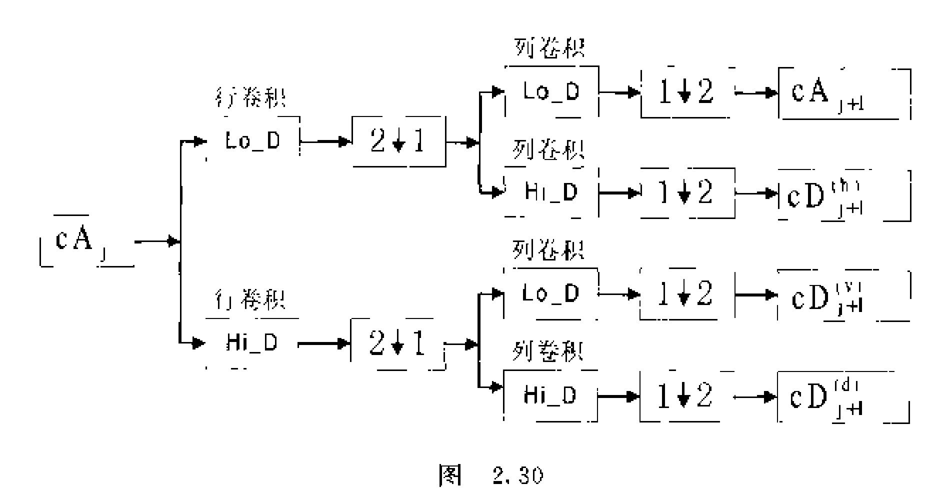


图　2.30

其中，

[2↓1] 表示列抽样：保留所有偶数列；

[1↓2] 表示行抽样：保留所有偶数行；

行卷积 [X] 与滤波器 X 进行行卷积运算，X 可以为 Lo_D 或 Hi_D；

列卷积 [X] 与滤波器 X 进行列卷积运算，X 可以为 Lo_D 或 Hi_D。

需要对该算法指出的是，为了对信号的边缘效果有较好的处理，要对原始信号进行拓展(用dwtmode函数控制一个全局变量实现，参见dwtmode)。拓展有三种方法：(1)补0拓展；(2)对称拓展；(3)平滑拓展。对于这三个拓展方法，dwt2有它唯一的重构函数，即idwt2。为了更加清楚地描述dwt2分解的过程，我们对第一步分解过程用图2.31描述：

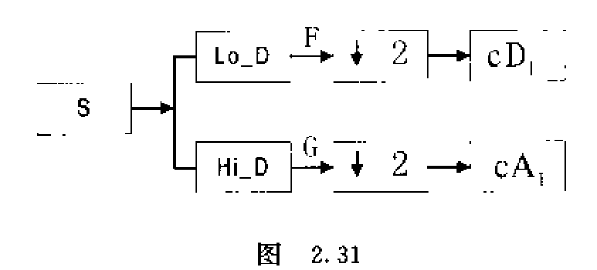


图　2.31

其中，

X	表示信号与滤波器的卷积；
↓2	表示抽取向量中标号为偶数的元素。

每个滤波器的长度均为2N，如果n=length(s)，则信号F和G的长度为n+2N−1，所以系数cA_1和cD_1的长度为$\text{floor}(\frac{n-1}{2})+N$。

举例：

```
%下面装载原始图像，X中含有被装载的信号，map中含有被装载的colormap
load woman;
nbcol=size(map,1);
%画出原始的图像
figure(1);
subplot(221); image(X); colormap(map);
title('原始图像');
axis square
%用小波函数db1对X进行单尺度分解
[cA,cH,cV,cD]=dwt2(X,'db1');
%画出各分解系数对应的图像
figure(2);
subplot(221); image(cA); colormap(map);
title('低频系数图像');
axis square
subplot(222); image(cH); colormap(map);
title('水平高频图像');
axis square
subplot(223); image(cV); colormap(map);
title('垂直高频图像');
axis square
subplot(224); image(cD); colormap(map);
```

```
title('斜线高频图像');
axis square
%从 figure(2)中画出的图可以看出，直接用系数画出的图像是很不清晰的，我们
%可以通过量化编码的手段改善图像的质量
%下面进行图像的编码
cod_X=wcodemat(X,nbcol);
cod_cA=wcodemat(cA,nbcol);
cod_cH=wcodemat(cH,nbcol);
cod_cV=wcodemat(cV,nbcol);
cod_cD=wcodemat(cD,nbcol);
figure(3);
subplot(221); image(cod_cA); colormap(map);
title('编码后低频系数图像');
axis square
subplot(222); image(cod_cH); colormap(map);
title('编码后水平高频图像');
axis square
subplot(223); image(cod_cV); colormap(map);
title('编码后垂直高频图像');
axis square
subplot(224); image(cod_cD); colormap(map);
title('编码后斜线高频图像');
axis square
```

输出结果(如图 2.32、2.33、2.34 所示)。

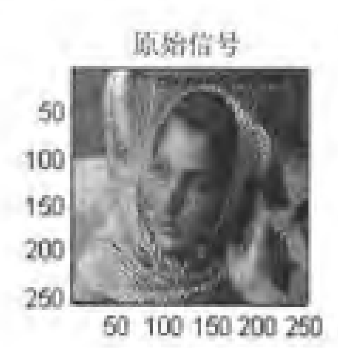


图　2.32

参见：dwtmode，dwtper，idwt，wavedec，waveinfo

2. idwt2

功能：单尺度逆二维离散小波变换

格式：① X=idwt2(cA,cH,cV,cD,'wname')

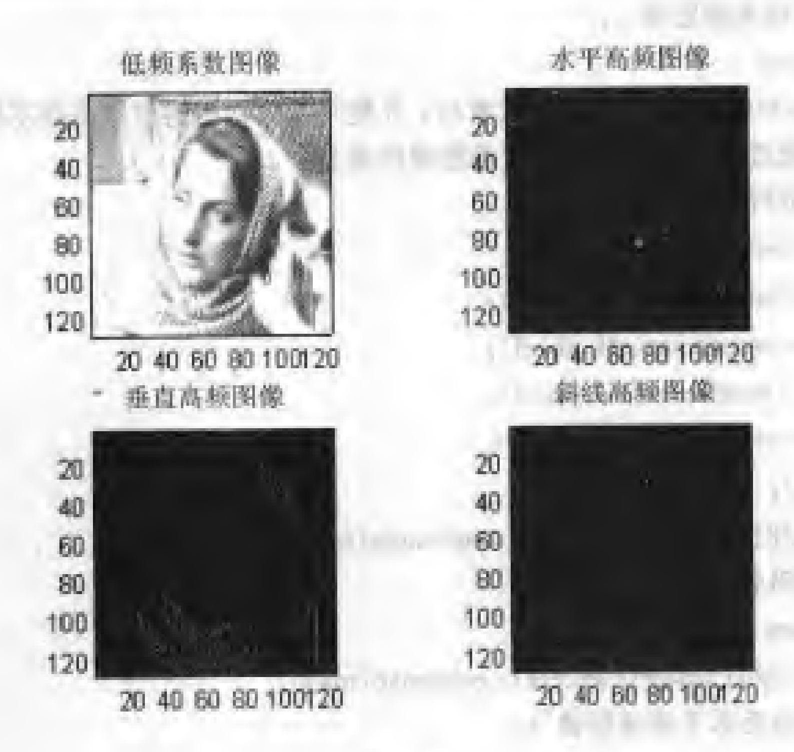


图　2.33

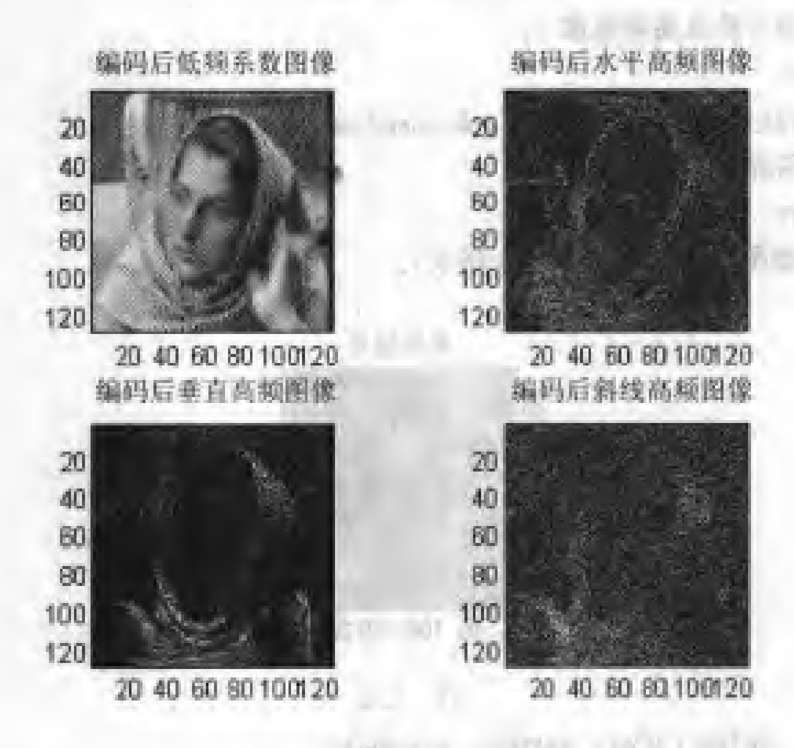


图　2.34

② X=idwt2(cA,cH,cV,cD,Lo_R,Hi_R)

③ X=idwt2(cA,cH,cV,cD'wname',S)

④ X=idwt2(cA,cH,cV,cD,Lo_R,Hi_R,S)

说明：该函数用于单尺度二维离散小波变换的重构，它通常和 dwt2 配套使用。对格式①、

③，它是用小波函数进行重构，对于②、④它是用重构滤波器进行重构，其中，Lo_R 和 Hi_R 的长度是相等的。返回向量 X 为单尺度重构后信号的低频系数。如果 size(cA)=size(cH)=size(cV)= size(cD)且滤波器的长度为 lf，则 X 的长度为 size(X)=2 * size(cA)−lf+2。对于格式③、④，则是返回用格式①、②对信号重构后的中间长度为 L 的部分，S 必须满足 S<2 * size(cA) −lf+2。

运用重构滤波器进行二维离散小波重构可以用图 2.35 进行描述。

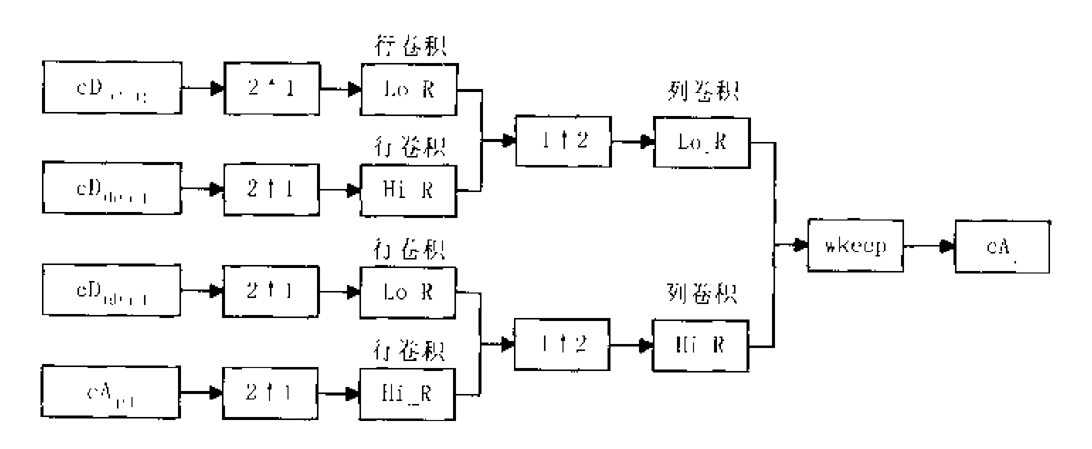


图　2.35

图中各符号的意义：

cA_j ——第 j 层的低频系数

cA_{j+1} ——第 j+1 层的低频系数

$cH_{j+1}^{(h)}$ ——第 j+1 层的水平高频系数

$cV_{j+1}^{(v)}$ ——第 j+1 层的垂直高频系数

$cD_{j+1}^{(d)}$ ——第 j+1 层的对角线高频系数

[2↑1] 表示列插样：在奇数列插入 0 值；

[1↑2] 表示行插样：在奇数行插入 0 值；

行卷积 [X] 与滤波器 X 进行行卷积运算，X 可以为 Lo_R 或 Hi_R；

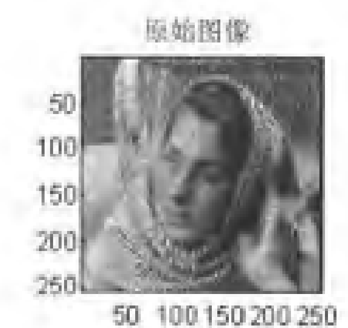
列卷积 [X] 与滤波器 X 进行列卷积运算，X 可以为 Lo_D 或 Hi_D。

举例：%下面装载原始图像，X 中含有被装载的信号，map 中含有被装载的 colormap

```
load woman;
sX=size(X);
%画出原始图像
subplot(221); image(X); colormap(map);
title('原始图像');
axis square
```

```
%用 db4 小波对 X 进行单尺度分解
[cA,cH,cV,cD]=dwt2(X,'db4');
lxtp=2 * size(cA)
%利用分解系数进行直接重构
A0=idwt2(cA,cH,cV,cD,'db4',sX);
%画出重构图像
subplot(222); image(A0); colormap(map);
title('重构图像');
axis square
%检查重构精度
A0max=max(max(X-A0))
```

输出结果(如图 2.36 所示):



重构图像

图 2.36

```
lxtp =
    262   262
A0max =
    3.3069e-010
```

参见：dwt2，idwtper2，upwlev2

3. detper2

功能：单尺度二维离散小波变换(周期性)

格式：① [cA,cH,cV,cD]=dwtper2(X, 'wname')

　　　② [cA,cH,cV,cD]=dwtper2(X, Lo_D, Hi_D)

说明：该函数是一个二维小波分析函数，它用来计算信号 X 的周期小波分解的低频系数矩阵 cA 和高频系数矩阵 cH、cV 和 cD。在分解时可以采用小波函数进行分解(如格式①)，也可以采用分解滤波器进行分解(如格式②)。cA、cH、cV 和 cD 矩阵的维数相等，如果 sx=size(X)，则 size(cA)=size(cH)=size(cV)=size(cD)=ceil(sx/2)。

举例：%下面装载原始图像，X 中含有被装载的信号，map 中含有被装载的 colormap

```
load woman;
nbcol=size(map,1);
```

```
%画出原始的图像
figure(1);
subplot(221); image(X); colormap(map);
title('原始图像');
axis square
%用小波函数 db1 对 X 进行单尺度分解
[cA,cH,cV,cD]=dwtper2(X,'db1');
%画出各分解系数对应的图像
figure(2);
subplot(221); image(cA); colormap(map);
title('低频系数图像');
axis square
subplot(222); image(cH); colormap(map);
title('水平高频图像');
axis square
subplot(223); image(cV); colormap(map);
title('垂直高频图像');
axis square
subplot(224); image(cD); colormap(map);
title('斜线高频图像');
axis square
%从 figure(2)中画出的图可以看出，直接用系数画出的图像是很不清晰的，我们
%可以通过量化编码的手段改善图像的质量
%下面进行图像的编码
cod_X=wcodemat(X,nbcol);
cod_cA=wcodemat(cA,nbcol);
cod_cH=wcodemat(cH,nbcol);
cod_cV=wcodemat(cV,nbcol);
cod_cD=wcodemat(cD,nbcol);
figure(3);
subplot(221); image(cod_cA); colormap(map);
title('编码后低频系数图像');
axis square
subplot(222); image(cod_cH); colormap(map);
title('编码后水平高频图像');
axis square
subplot(223); image(cod_cV); colormap(map);
title('编码后垂直高频图像');
axis square
```

```
subplot(224); image(cod_cD); colormap(map);
title('编码后斜线高频图像');
axis square
```

输出结果(如图2.37、2.38、2.39所示)。

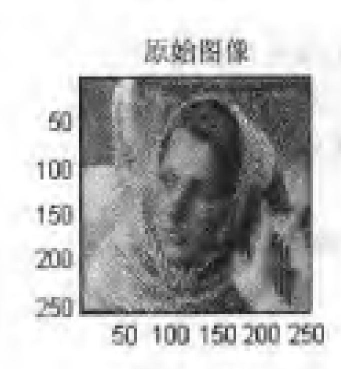


图 2.37

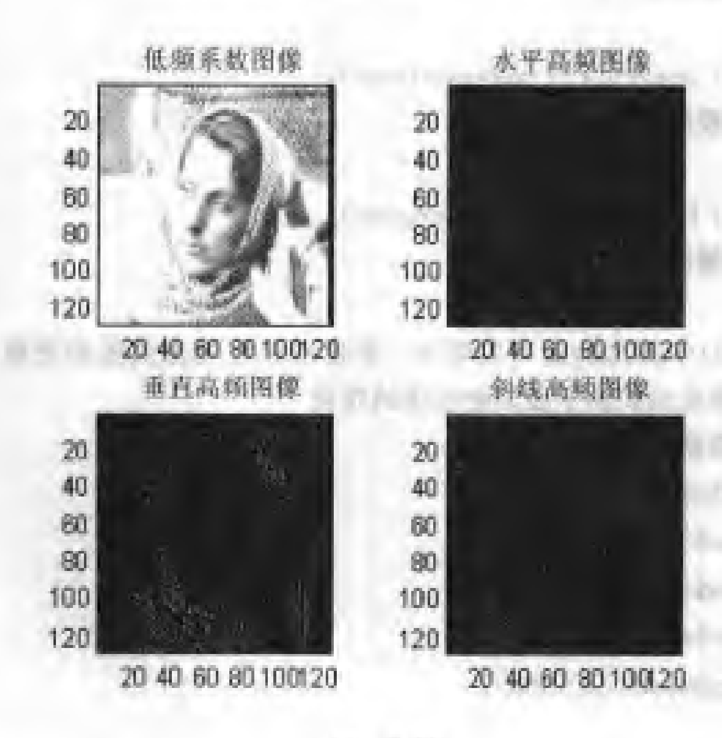


图 2.38

参见: dwt2, idwtper2

4. idwtper2

功能: 单尺度二维离散逆小波分析(周期性)

格式: ① X=idwtper2(cA,cH,cV,cD,'wname')

② X=idwtper2(cA,cH,cV,cD,Lo_R,Hi_R)

③ X=idwtper2(cA,cH,cV,cD,'wname',S)

④ X=idwtper(cA,cH,cV,cD,Lo_R,Hi_R,S)

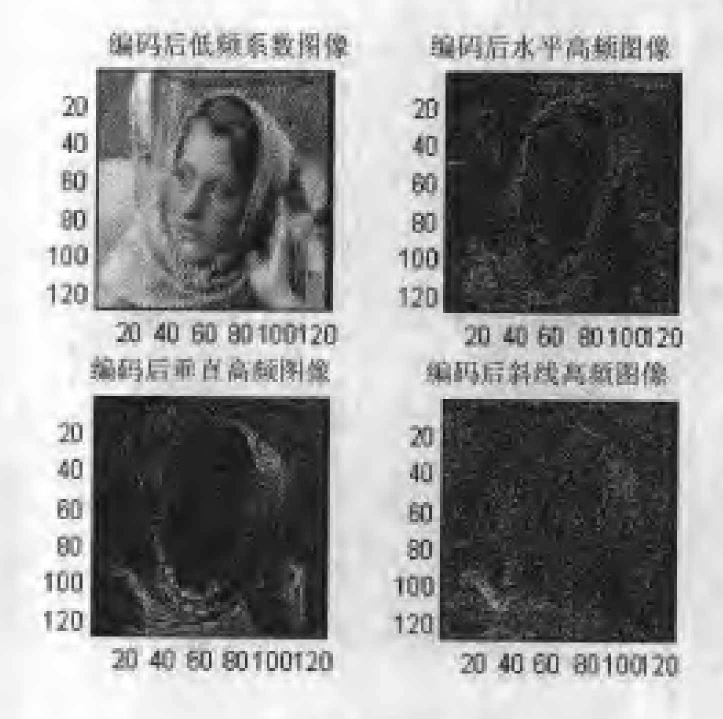


图　2.39

说明：该函数用于单尺度二维离散小波分解(周期性)的重构，它是 dwtper2 的逆函数，即有 idwtper2(dwtper2(X,'wname')=X)。对格式①、③，它是用小波函数进行重构，对于②、④它是用重构滤波器进行重构，其中，Lo_R 的 Hi_R 的长度是相等的。返回系数 X 为重构后信号的向量。如果 sa=size(cA)=size(cH)=size(cV)=size(cD)，则 size(X)=2*sa。对于格式③、④，则是返回用格式①、②对信号重构后的中间长度为 S 的部分。

举例：%下面装载原始图像，X 中含有被装载的信号，map 中含有被装载的 colormap

```
load woman;
sX=size(X);
%画出原始图像
subplot(221); image(X); colormap(map);
title('原始图像');
axis square
%用 db4 小波对 X 进行单尺度周期性分解
[cA,cH,cV,cD]=dwtper2(X,'db4');
lxtp=2*size(cA)
%利用分解系数进行直接重构
A0=idwtper2(cA,cH,cV,cD,'db4',sX);
%画出重构图像
subplot(222); image(A0); colormap(map);
title('重构图像');
axis square
%检查重构精度
```

```
A0max=max(max(X-A0))
```

输出结果(如图 2.40 所示):

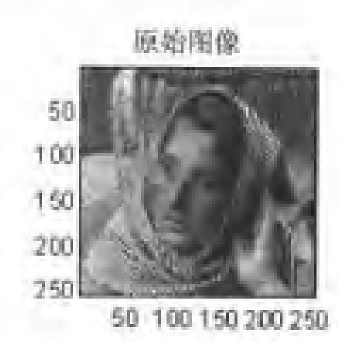


图 2.40

```
lxtp =
     256    256
A0max =
     3.3069e-010
```

参见:dwtper2

5. wavedec2

功能:多尺度二维小波分解(二维多分辨分析函数)

格式:① [C,S]=wavedec2(X, N, 'wname')

　　② [C,S]=wavedec2(X, N, Lo_D, Hi_D)

说明:wavedec2 函数完成对信号 X 在尺度 N 上的二维分解,N 是严格的正整数。格式①用小波进行二维分解;格式②用低通分解滤波器(Lo_D)和高通分解滤波器(Hi_D)进行二维分解。返回参数为分解结构[C, S],其中,C 为

$$C=[A(N)\mid H(N)\mid V(N)\mid D(N)\mid \cdots H(N-1)\mid V(N-1)\mid D(N-1)\mid \cdots H(1)\mid V(1)\mid D(1)]$$

向量 A 为低频系数,向量 H 为水平高频系数,向量 V 为垂直高频系数,向量 D 为斜线高频系数。每个向量是以列的方向存储在矩阵 C 中。

矩阵 S 为

S(1,:)为尺度 N 的低频系数的长度。

S(i,:)为尺度 N−i+2 的高频系数的长度,其中,i=2,…,N+1。

S(N+2,:)=size(X)

对于 wavedec2 的分解过程及分解后的形式可以大致如图 2.41 所示。

下面对算法进行说明:对于图像分解,其算法同一维分解的情况类似,二维小波函数和尺度函数是通过一维小波函数和尺度函数经过张量积(tensor product)变换得到。二维小波分解是把尺度 j 的低频部分分解成四个部分:尺度 j+1 的低频部分和三个方向(水平、垂直、斜线)的高频部分。其基本的分解步骤可用图 2.42 表示。

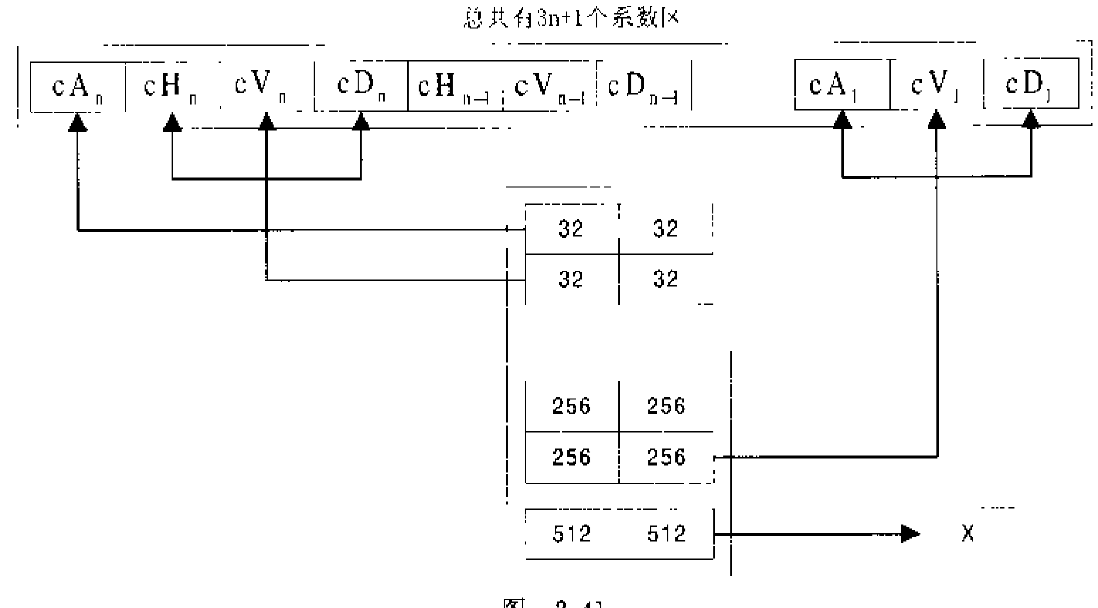


图　2.41

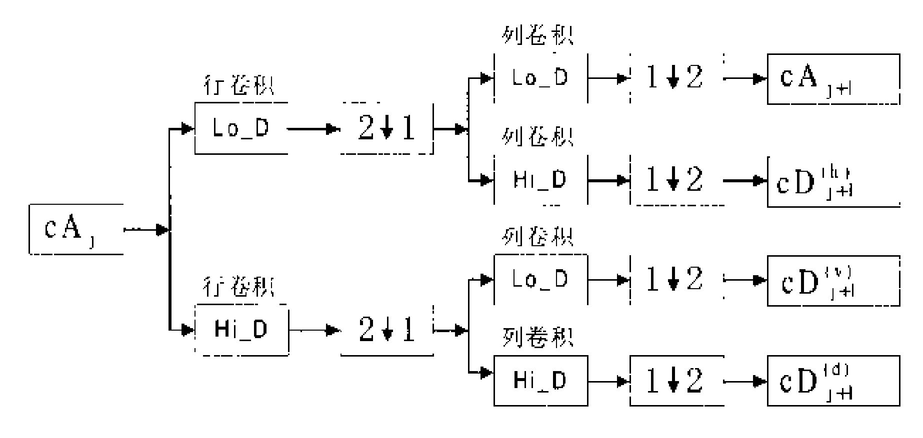


图　2.42

其中，

2↓1	表示列抽样：保留所有偶数列；
1↓2	表示行抽样：保留所有偶数行；
行卷积 X	与滤波器 X 进行行卷积运算，X 可以为 Lo_D 或 Hi_D；
列卷积 X	与滤波器 X 进行列卷积运算，X 可以为 Lo_D 或 Hi_D。

举例：%下面装载原始图像，X 中含有被装载的信号

```
load woman;
%用小波函数 db1 对 X 进行尺度 2 的分解
[c,s]=wavedec2(X,2,'db1');
sizex=size(X)
```

```
sc=size(c)
val_s=s
```

输出结果：

```
sizex =
      256     256
sc =
                   1     65536
val_s =
        64      64
        64      64
       128     128
       256     256
```

参见：dwt，waveinfo，wfilters，wmaxlev

6. waverec2

功能：多尺度二维小波重构

格式：① X=waverec2(C,S,'wname')

② X=waverec2(C,S,Lo_R,Hi_R)

说明：该函数是用指定的小波函数或重构滤波器在小波分解结构[C，S]上对信号X进行多尺度二维小波重构，它是wavedec2函数的逆函数，即有 X=waverec2(wavedec2(X,N,'wname'),'wname')。另外，X=waverec2(C,S,'wname')与X=appcoef2(C,S,'wname',0)具有等价性。

举例：

```
%下面装载原始图像，X中含有被装载的信号
load woman;
%画出原始图像
subplot(221); image(X); colormap(map);
title('原始图像');
axis square
%用小波函数sym4对X进行2层小波分解
[c,s]=wavedec2(X,2,'sym4');
%对小波分解结构[c,s]进行重构
a0=waverec2(c,s,'sym4');
%画出重构图像
subplot(222); image(X); colormap(map);
title('重构图像');
axis square
%检查X的最大重构误差
a0max=max(max(X-a0))
```

输出结果(如图2.43所示)：



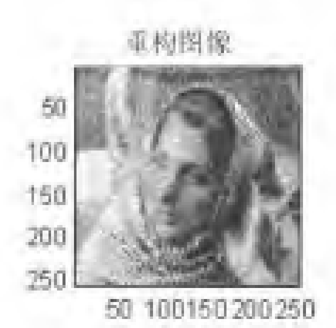


图　2.43

```
a0max =
    2.0033e-010
```

参见：appcoef2，idwt2，wavedec2

7. appcoef2

功能：提取二维小波分解低频系数

格式：① A=appcoef2(C，S，'wname'，N)

② A=appcoef2(C，S，'wname')

③ A=appcoef2(C，S，Lo_R，Hi_R)

④ A=appcoef2(C，S，Lo_R，Hi_R，N)

说明：该函数是一个二维小波分析函数，它主要用于从多尺度二维小波分解的分解结构[C,S](参见 wavedec2) 中提取二维信号的低频系数，从某种意义上说，它常常和 wavedec2 函数配套使用。格式①计算尺度为 N(N 必须为一个正整数且 0≤N≤length(S)－2)，小波函数为 wname，分解结构为[C，S]时的二维分解低频系数；格式②用于提取最后一尺度(N＝length(S)－2)的小波变换低频系数；格式③、④是用重构滤波器 Lo_R 和 Hi_R 进行信号低频系数的提取。

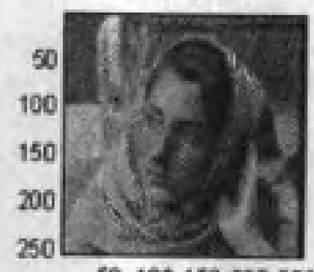
举例：%下面装载二维信号，X 含有被装载的信息

```
load woman;
%画出原始图像
subplot(221); image(X); colormap(map);
title('原始图像');
axis square
%尺度为 2，小波函数为 db1 时进行分解
[c,s]=wavedec2(X,2,'db1');
sizex=size(X)
sizec=size(c)
val_s=s
%提取尺度 2 中的低频信号
ca2=appcoef2(c,s,'db1',2);
```

```
sizeca2=size(ca2)
%画出尺度2中低频信号的图像
subplot(222)；image(ca2)；colormap(map)；
title('尺度2的低频图像')；
axis square
%提取尺度1中的低频系数
ca1=appcoef2(c,s,'db1',1)；
sizeca1=size(ca1)
subplot(223)；image(ca1)；colormap(map)；
title('尺度1的低频图像')；
axis square
```

输出结果(如图2.44所示)：





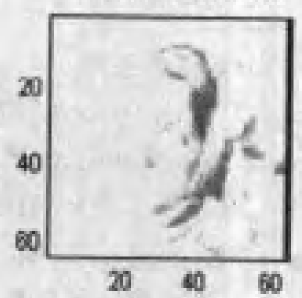




图 2.44

```
sizex =
    256    256
sizec =
           1      65536
val_s =
     64     64
     64     64
    128    128
    256    256
sizeca2 =
```

```
      64      64
  sizeca1 =
     128     128
```

参见：detcoef，wavedec

8. detcoef2

功能：提取二维小波分解高频系数

格式：D=detcoef2(type,C，S，N)

说明：该函数是一个二维小波分析函数，它与 appcoef2 函数相对应，用来从分解结构[C，S]中提取二维小波变换的高频系数，从某种意义上说，它也是常常和 wavedec2 函数配套使用。尺度 N 必须为一个正整数且 1⩽N⩽size(S,1)−2，type 为提取类型，其具体意义见表 2-3。该函数返回一个向量，且其长度为 length(S)/2^N。

表 2-3　type 的类型

type 的类型	意　　义
h	提取水平系数
v	提取垂直系数
d	提取对角线系数

举例：

```
%下面装载二维信号，X 含有被装载的信息
load woman;
%画出原始图像
figure(1);
subplot(221); image(X); colormap(map);
title('原始图像');
axis square
%尺度为 2，小波函数为 db1 时进行分解
[c,s]=wavedec2(X,2,'db1');
sizex=size(X)
sizec=size(c)
val_s=s
%从不同的方位从分解结构[c,s]中提取尺度 2 的高频系数
chd2=detcoef2('h',c,s,2);
cvd2=detcoef2('v',c,s,2);
cdd2=detcoef2('d',c,s,2);
sizecd2=size(chd2)
%画出尺度 2 高频信号的图像
subplot(2,2,2); image(chd2);
title('高频的水平部分 chd2')
```

```
axis square
subplot(2,2,3); image(cvd2);
title('高频的垂直部分 cvd2')
axis square
subplot(2,2,4); image(cdd2);
title('高频的斜线部分 cdd2')
axis square
%从不同的方位从分解结构[c,s]中提取尺度1的高频系数
chd1=detcoef2('h',c,s,1);
cvd1=detcoef2('v',c,s,1);
cdd1=detcoef2('d',c,s,1);
sizecd1=size(chd1)
%画出尺度1高频信号的图像
figure(2);
subplot(2,2,1); image(X); colormap(map);
title('原始图像');
axis square
subplot(2,2,2); image(chd1);
title('高频的水平部分 chd1')
axis square
subplot(2,2,3); image(cvd1);
title('高频的垂直部分 cvd1')
axis square
subplot(2,2,4); image(cdd1);
title('高频的斜线部分 cdd1')
axis square
```

输出结果(如图 2.45、2.46 所示)。

```
sizex =
    256    256
sizec =
              1      65536
val_s =
     64     64
     64     64
    128    128
    256    256
sizecd2 =
     64     64
sizecd1 =
```

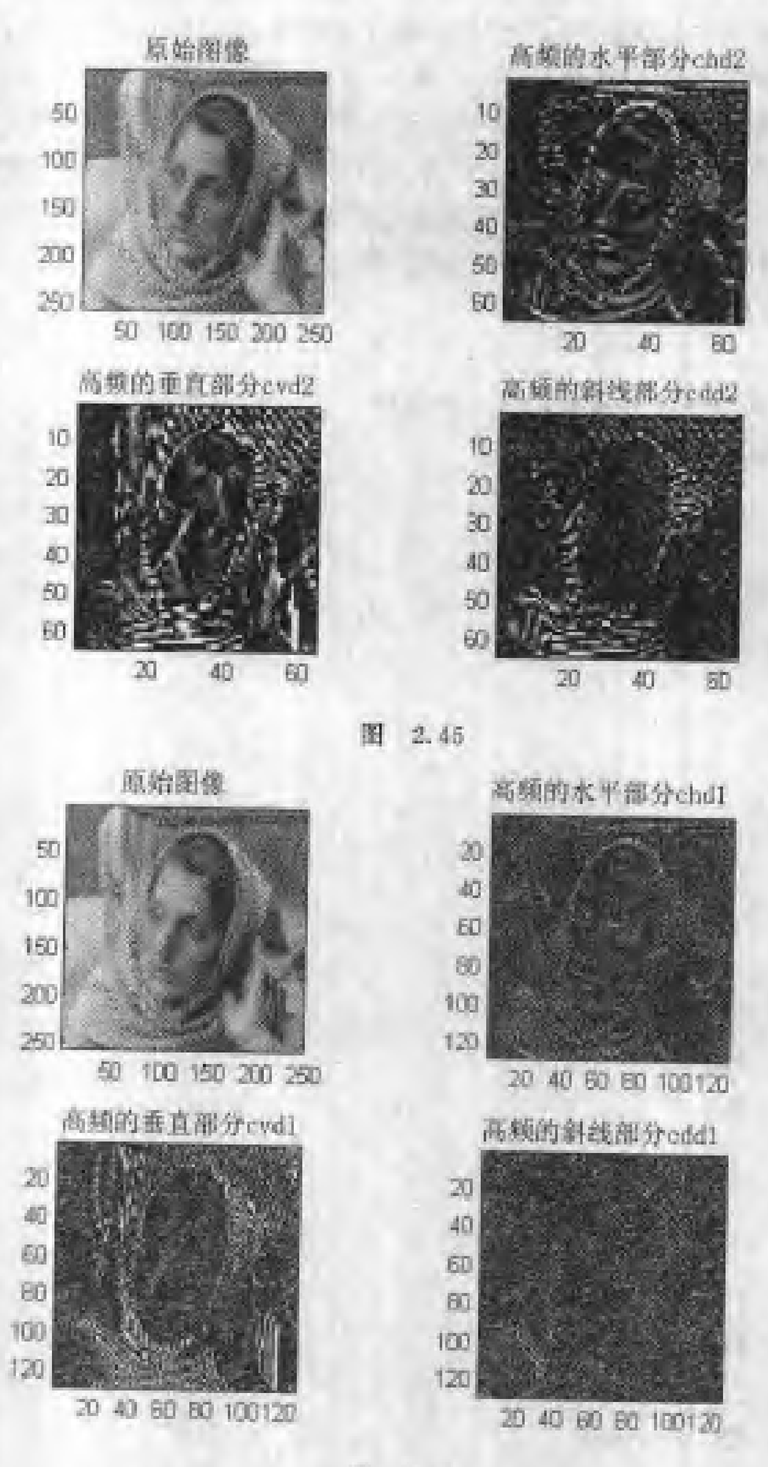


图　2.45

图　2.46

128　　128

参见：appcoef，wavedec

9. upwlev2

功能：二维小波分解的单尺度重构

格式：① [NC,NL,cA]=upwlev2(C,S,'wname')

② [NC,NL,cA]=upwlev2(C,S,Lo_R,Hi_R)

说明：该函数用于对小波分解结构[C,S]进行单尺度重构，即对分解结构[C,S]的第 n 步进行重构，返回一个新的分解结构[NC,NL](第 n－1 步的分解结构)，并提取和最后一尺度的低频系数矩阵。即如果[C,S]是尺度 n 的一个分解结构，则[NC,NS]是尺度 n－1 的一个分解结构，cA 是尺度 n 的低频系数矩阵。C 为原始的小波分解向量，S 是相应的记录矩阵。

对于格式②，函数是用低通滤波器和高通滤波器进行重构。

举例：

```
%下面装载原始图像，X 中含有被装载的信号
load woman;
sX=size(X);
%用小波函数 db1 对 X 进行尺度 2 的分解
[c,s]=wavedec2(X,2,'db1');
sc=size(c)
val_s=s
%对小波分解结果进行一步重构
[nc,ns]=upwlev2(c,s,'db1');
snc=size(nc)
val_ns=ns
```

输出结果：

```
sc =
            1        65536
val_s =
      64     64
      64     64
     128    128
     256    256
snc =
            1        65536
val_ns =
     128    128
     128    128
     256    256
```

参见：idwt2，upcoef2，wavedec2

10. wrcoef2

功能：对二维小波系数进行单支重构

格式：① X=wrcoef2('type',C,S,'wname',N)

② X=wrcoef2('type',C,S,Lo_R,Hi_R,N)

③ X=wrcoef2('type',C,S,'wname')

④ X=wrcoef2('type',C,S,Lo_R,Hi_R)

说明：该函数是对二维信号的分解结构[C, S]用指定的小波函数或重构滤波器进行重构。

当 type='a'时，指对信号的低频部分进行重构，此时 N 可以为 0；当 type='h'(或'v'、'd')时，指对信号水平(或垂直、对角线(或称斜线))的高频部分进行重构。N 为严格的正整数且有

当 type=a 时，0≤N≤size(S,1)-2

当 type=h、v 或 d 时，1≤N≤size(S,1)-2

对于格式①、③，函数用小波函数进行重构，对于格式②、④，函数用重构滤波器进行重构。对于格式③、④，重构系数的最大尺度为 N=size(S,1)-2。

举例：%下面装载原始图像，X 中含有被装载的信号

```
load woman;
%画出原始图像
subplot(221); image(X); colormap(map);
title('原始图像');
axis square
%用小波函数 sym5 对 X 进行尺度 2 的分解
[c,s]=wavedec2(X,2,'sym5');
%对小波分解结构[c,s]的低频系数分别进行尺度 1 和 2 上的重构
a1=wrcoef2('a',c,s,'sym5',1);
a2=wrcoef2('a',c,s,'sym5',2);
%画出尺度 1 的低频图像
subplot(222); image(a1); colormap(map);
title('尺度 1 的低频图像');
axis square
%画出尺度 2 的低频图像
subplot(223); image(a2); colormap(map);
title('尺度 2 的低频图像');
axis square
%对小波分解结构[c,s]的高频系数分别进行尺度 2 上的重构
hd2=wrcoef2('h',c,s,'sym5',2);
vd2=wrcoef2('v',c,s,'sym5',2);
dd2=wrcoef2('d',c,s,'sym5',2);
%画出尺度 2 水平高频图像，其它的情况与此完全一样
subplot(224); image(hd2); colormap(map);
title('尺度 2 水平高频图像');
axis square
%检查重构图形的大小
sX=size(X) %原始图像
sa1=size(a1) %低频重构图像
shd2=size(hd2) %高频重构图像
```

输出结果(如图 2.47 所示)：

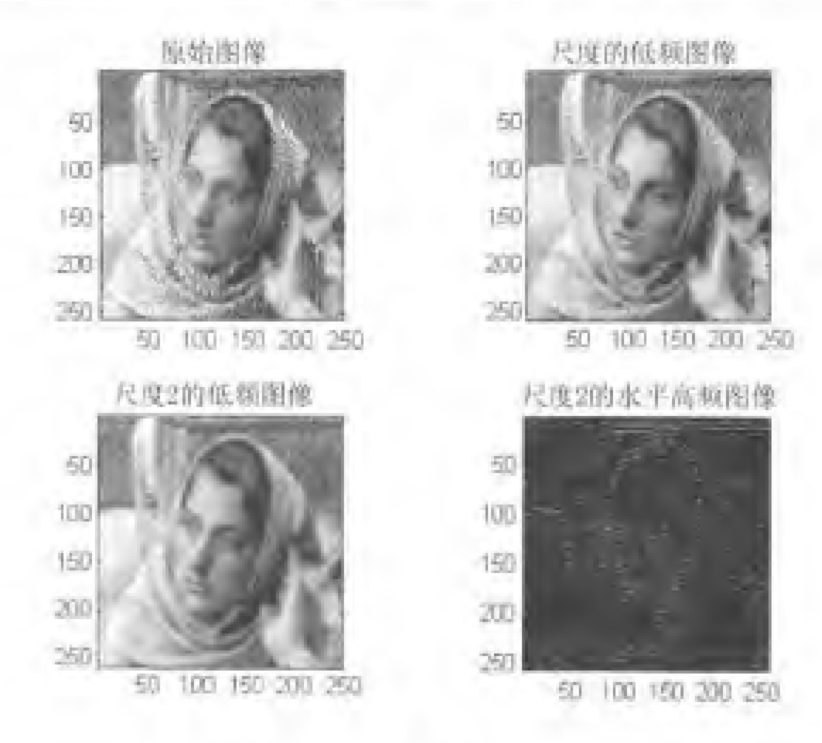


图　2.47

```
sX =
    256    256
sa1 =
    256    256
shd2 =
    256    256
```

参见：appcoef2，detcoef2，wavedec2

11. upcoef2

功能：二维小波分解的直接重构

格式：① Y=upcoef2(O,X,'wname',N,S)

② Y=upcoef2(O,X,Lo_R,Hi_R,N,S)

③ Y=upcoef2(O,X,'wname',N)

④ Y=upcoef2(O,X,Lo_R,Hi_R,N)

⑤ Y=upcoef2(O,X,'wname')

⑥ Y=upcoef2(O,X,Lo_R,Hi_R)

说明：该函数是用于二维小波分析的函数，它是对第N层的小波分解系数进行重构，N是严格的正整数。如果O='a'，则是对低频系数进行重构，如果O='h'(或v，d)，则是对水平方向(或垂直方向，对角线(斜线)方向)的高频系数进行重构。对于格式①、②的情况，则是对向量X进行重构并返回中间长度为S的部分。在重构的过程中，格式①、③、⑤是用小波函数进行重构，格式②、④、⑥是用重构滤波器进行重构。

另外需要注意的是，Y=upcoef2(O,X,'wname')等价于Y=upcoef2(O,X,'wname',1)；

Y=upcoef2(0,X,Lo_R,Hi_R)等价于 Y=upcoef2(0,X,Lo_R,Hi_R, 1)。

举例：

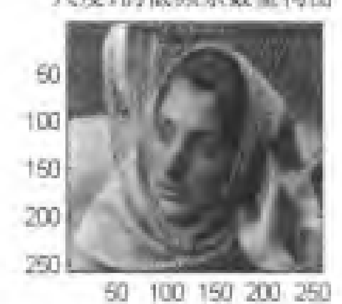
```
%下面装载原始图像，X 中含有被装载的信号
load woman;
sX=size(X);
%画出原始图像
figure(1);
subplot(221); image(X); colormap(map);
title('原始图像');
axis square
%用小波函数 db4 对 X 进行尺度 2 的分解
[c,s]=wavedec2(X,2,'db4');
%从分解系数中对尺度 1 的低频部分和高频部分进行重构，
%我们可以用函数 wrcoef2 完成，也可以用下面的过程等价地完成
%第一步：从小波分解结构中提取系数；
%第二步：用 upcoef2 函数进行重构
siz=s(size(s,1),:);
%提取低频部分系数并进行重构
ca1=appcoef2(c,s,'db4',1,siz);
a1=upcoef2('a',ca1,'db4',1,siz);
%画出低频重构图
figure(2);
subplot(221); image(a1); colormap(map);
title('尺度 1 的低频系数重构图');
axis square
%提取高频水平部分系数并进行重构
chd1=detcoef2('h',c,s,1);
hd1=upcoef2('h',chd1,'db4',1,siz);
%画出高水平直部分重构图
subplot(222); image(hd1); colormap(map);
title('尺度 1 的高频水平重构图');
axis square
%提取高频垂直部分系数并进行重构
cvd1=detcoef2('v',c,s,1);
vd1=upcoef2('v',cvd1,'dh4',1,siz);
%画出高频垂直部分重构图
subplot(223); image(vd1); colormap(map);
title('尺度 1 的高频垂直重构图');
axis square
%提取高频斜线部分系数并进行重构
```

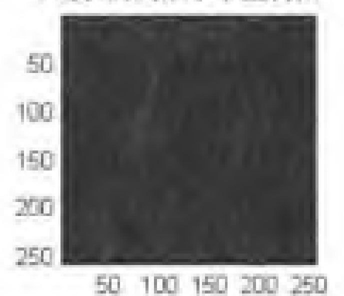
```
cdd1=detcoef2('d',c,s,1);
dd1=upcoef2('d',cdd1,'db4',1,siz);
%画出高频斜线部分重构图
subplot(224); image(dd1); colormap(map);
title('尺度 1 的高频斜线重构图');
axis square
```

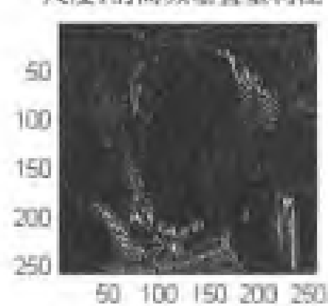
输出结果(如图 2.48、2.49 所示):



图 2.48







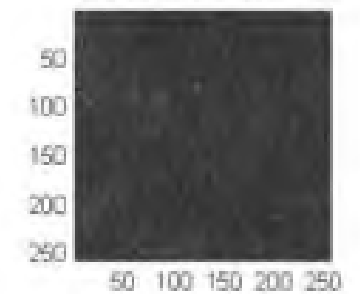


图 2.49

参见: idwt2

2.5 小 波 包 算 法

在这一节里，我们将详细讲述关于小波包分析方面一些函数的用法。它们的主要功能有：一维和二维小波包分解、一维和二维小波包重构、最佳(优)小波包基的选择(即最优树的选择)等。

1. wpdec

功能：一维小波包的分解

格式：① [T,D]=wpdec(X,N,'wname',E,P)

② [T,D,RN]=wpdec(X,N,'wname')

说明：wpdec 是一个一维小波包分解函数。

对于格式①，它根据小波函数'wname'(参见 wfilters)、熵标准 E 和参数 P 对信号 X 进行 N 层小波包分解，并返回小波包分解结构[T,D](T 为树结构，D 为数据结构)。其中，E 是用来指定熵标准(参见 wentropy)，E 的类型可以有：'shannon'、'threshold'、'norm'、'log energy'、'sure'或'user'。P 是一个可选的参数，它的选择根据参数 E 的值来决定。

- 如果 E='shannon'或'log energy'时，则 P 不用；
- 如果 E='threshold' 或'sure'时，则 P 是阈值，并且必须为正数；
- 如果 E='norm'时，则 P 是指数，且有 $1 \leqslant P < 2$；
- 如果 E='user'时，则 P 是一个包含 *.M 文件名的字符串，这个 *.M 文件是在一个输入变量 X 下，用户自己的熵函数。

格式②等价于[T,D]=wpdec(X,N,'wname','shannon')，即默认为'shannon'熵标准。

小波包分析是多分辨分析的推广，它提供了更为丰富和精确的信号分析方法。小波包元素是由三个参数来确定的一个波形，这三个最基本参数是：位置、尺度(与一般小波分解一样)和频率。

对于一个给定的正交小波函数，可以生成一组小波包基，每一个小波包基提供一种特定的信号编码方法。它完整地保存了信号的全部能量，并可以对原始信号的全部特征进行完全重构。利用小波包，可以对信号进行大量不同方式的分解，但对于给定的信号 X、小波包函数'wname'、分解层数 N 和熵标准 E，它具有一种最佳的小波包分解方式。

对小波包的分解和最佳分解方式的选择，都有着简单而有效的算法，在此我们介绍一种自适应滤波器算法，它可以直接用于最优信号的编码和数据压缩。

在正交小波分解过程中，一般的方法是将低频系数分解成两部分，分开后，我们获得一个新的低频系数和一个高频系数的向量，在两个连续低频系数中间丢失的信息被高频系数获得。下一步是将新的低频系数向量继续分解成两部分，而高频系数不会被再分解。

在小波包分解中，每一个高频系数向量也像低频部分的分解一样，被分解成两部分，因而，它提供了更丰富的分析方法。在一维情况下，它产生一个完整的二叉树；在二维情况下，它产生一个完整的四叉树。

举例：%装入信号

```
load noisdopp; x=noisdopp;
```

```
%用 db1 小波包分解信号 x 到第三层
%采用 shannon 熵的标准
[t,d]=wpdec(x,3,'db1','shannon');
plottree(t) %画树结构的图形
```

输出结果(如图 2.50 所示):

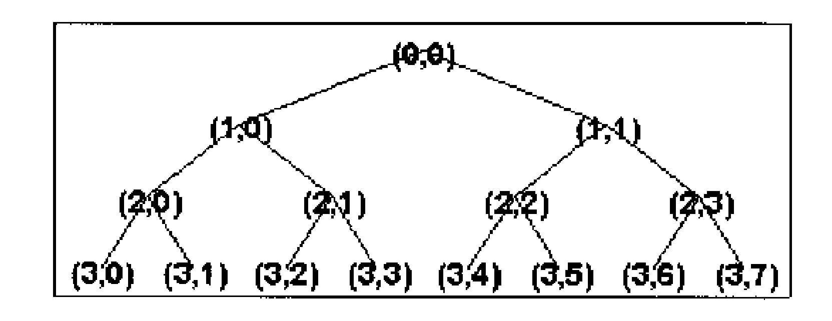


图 2.50

参见: maketree,waveinfo,wdatamgr,wentropy,wpdec2,wtreemgr

2. wprec

功能: 一维小波包分解的重构

格式: X=wprec(T,D)

说明: wprec 是一个一维小波包分析函数，它对小波包的分解结构[T,D]进行重构，并返回重构后的向量 X，其中 T 是树结构，D 是数据结构(参见 maketree)。因为有 X=wprec(wpdec(X,'wname'))，所以说，wprec 是 wpdec 的反函数。

举例: %装入信号

```
load noisdopp; x=noisdopp(1:1000);
figure(1);
subplot(211); plot(x);
title('原始信号');
%用 db1 小波包对信号 x 进行三层分解
[t,d]=wpdec(x,3,'db1','shannon');
plottree(t) %画小波包树结构的图形
recx=wprec(t,d); %重构小波包分解结构[T, D]
figure(1);
subplot(212); plot(recx);
title('重构后的信号');
```

输出结果(如图 2.51、2.52 所示)。

参见: maketree,wpdec,wpdec2,wpjoin,wprec2,wpsplt

3. wpdec2

功能: 二维小波包的分解

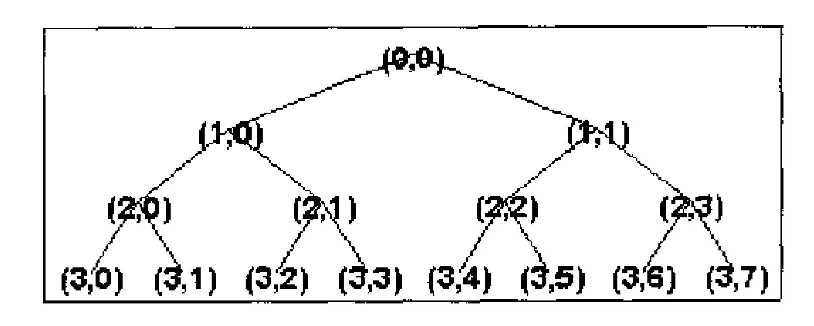
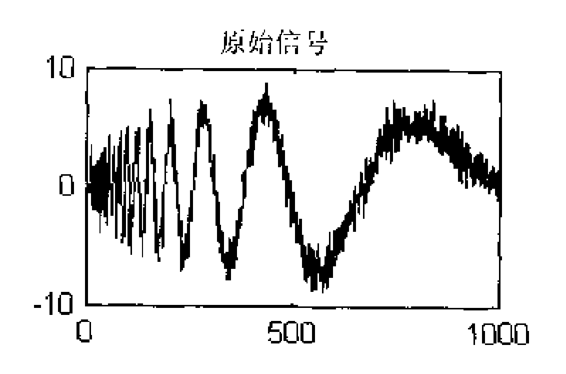


图 2.51



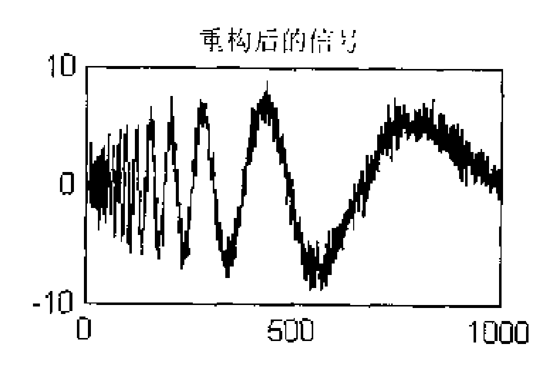


图 2.52

格式：① [T,D]=wpdec2(X,N,'wname',E,P)

　　　② [T,D,RN]=wpdec2(X,N,'wname')

说明：wpdec2 是一个二维小波包分析函数。格式①根据相应的小波包分解向量 X 和指定的小波('wname'，参见 wfilters)对 X 进行 N 层分解，并返回树结构 T 和数据结构 D。其中，E 是一个字符串，用来指定熵类型(参见 wentropy)，E 的类型可以有：'shannon'、'threshold'、'norm'、'log energy'、'sure'或'user'；P 是一个可选的参数，它根据参数 E 的值来确定。

如果 T='shannon'或'log energy'时，则 P 不用；

如果 T='threshold' 或'sure'时，则 P 是阈值，并且必须为正数；

如果 T='norm'时，则 P 是指数，且有 1≤P<2；

如果 T='user'时，则 P 是一个包含 *.M 文件名的字符串，这个 *.M 文件是在一个输入变量 X 下，用户自己的熵函数。

[T,D]=wpdec2(X,N,'wname')与[T,D]=wpdec2(X,N,'wname', 'shannon')是等价的。

举例：%装入图像，X 包含装入的图像

```
load tire;
%用默认的 shannon 熵，分解图像
[t,d]=wpdec2(X,2,'db1');
plottree(t) %画四叉树结构图
```

输出结果(如图 2.53 所示)。

参见：maketree,waveinfo,wdatamgr,wentropy,wpdec2,wtreemgr

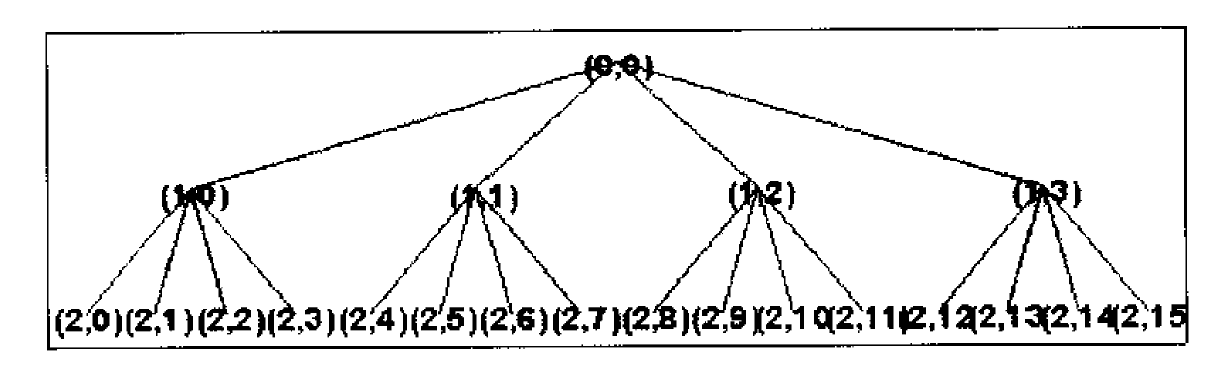


图　2.53

4. wprec2

功能：二维小波包分解的重构

格式：X＝wprec2(T,D)

说明：wprec2 是一个二维小波包分析函数，它对小波包的分解结构[T,D]进行重构，并返回重构后的向量 X，其中 T 是树结构，D 是数据结构(参见 maketree)。因为有 X＝wprec2(wpdec2(X,'wname'))，所以说，wprec2 是 wpdec2 的反函数。

举例：%装入图像，X 包含装入的图像

```
load tire;
figure(1);
subplot(221); image(X); colormap(map)
title('原始图像');
axis square
%用默认的 shannon 熵，分解图像
[t,d]=wpdec2(X,2,'db1');
plottree(t)      %画四阶树结构图
%对分解结构[t,d]进行重构
rectire=wprec2(t,d);
%画出重构后的图像
figure(1);
subplot(222); image(rectire); colormap(map)
title('重构后的图像');
axis square
```

输出结果(如图 2.54、2.55)。

参见：maketree,wpdec,wpdec2,wpjoin,wprec,wpsplt

5. wpcoef

功能：计算小波包系数

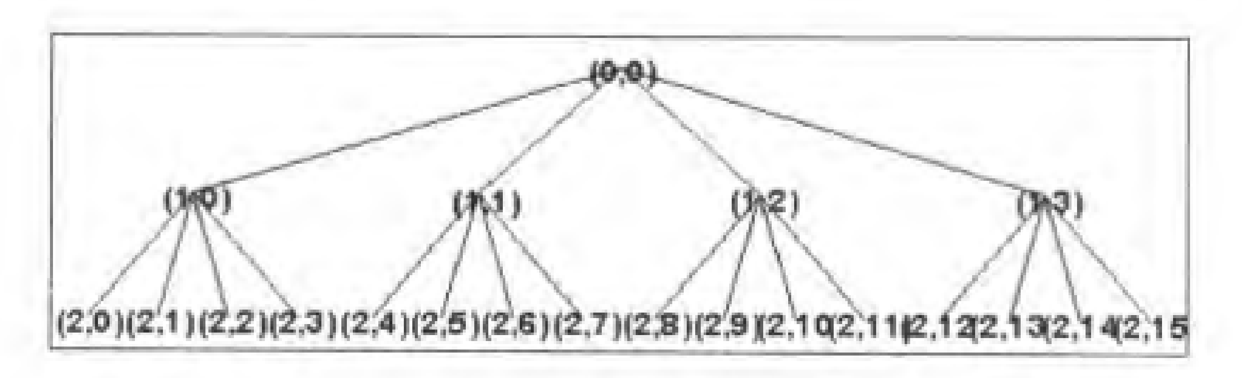


图 2.54

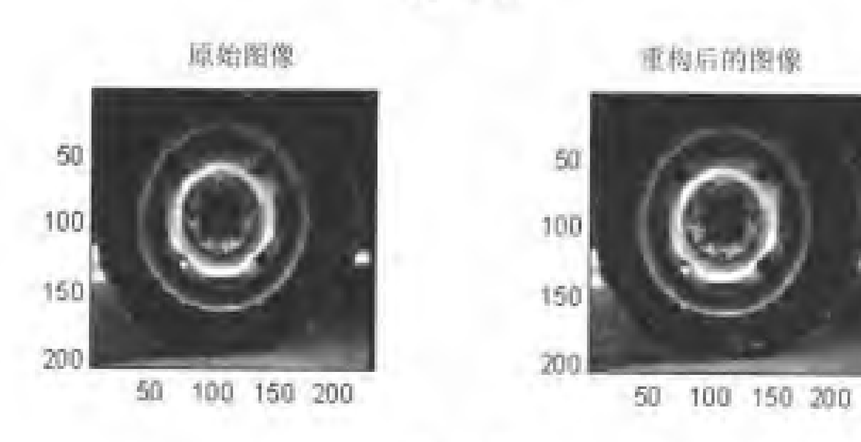


图 2.55

格式：① x=wpcoef(T,D,N)

② x=wpcoef(T,D)

说明：wpcoef 是一个一维或二维的小波包分析函数。格式①返回与结点 N 对应的系数，其中，T 是树结构，D 是数据结构。如果 N 不存在，则 x=[]。格式②等价于 x=wpcoef(T,D,0)。

举例：%装入信号

```
load noisdopp; x=noisdopp;
figure(1);
subplot(221); plot(x);
title('原始信号');
%用 db1 小波包分解信号 x 到第三层
[t,d]=wpdec(x,3,'db1','shannon');
plottree(t) %画树结构图形
%读取小波包(2,1)的系数
cfs=wpcoef(t,d,[2,1]);
figure(1);
subplot(222); plot(cfs);
title('小波包(2,1)的系数');
```

输出结果(如图 2.56、2.57 所示)：

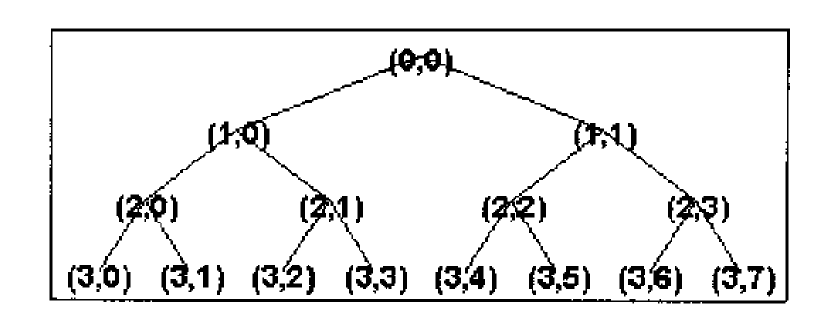
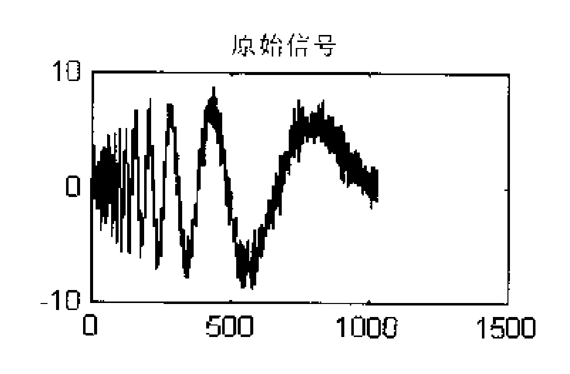


图 2.56



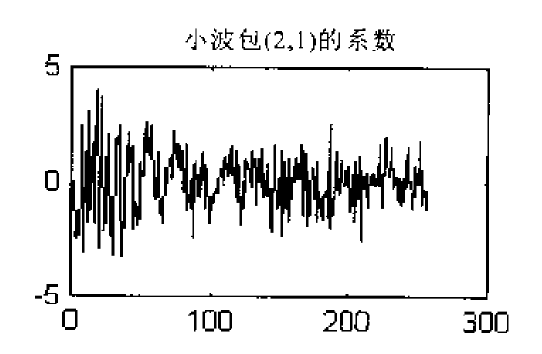


图 2.57

参见：maketree，wpdec，wpdec2

6. wprcoef

功能：小波包分解系数的重构

格式：X＝wprcoef(T，D，N)

说明：wprcoef 是一个一维或二维小波包分析函数，它计算结点 N 的小波包分解系数的重构信号(图像)，T 是树结构，D 是数据结构(参见 maketree)。X＝wprcoef(T，D)等价于 X＝wprcoef(T，D，0)，即完全重构原始信号。

另外需要强调一点，wprcoef 函数一次只能对一个结点进行重构，而不能同时对多个结点进行重构。如果想要对多个结点进行重构，需要多次调用该函数。

举例：%装入信号

```
load noisdopp；x＝noisdopp(1：1000)；
figure(1)；
subplot(211)；plot(x)；
title('原始信号')；
%用 db1 小波包对信号 x 进行三层分解
[t，d]＝wpdec(x，3，'db1'，'shannon')；
plottree(t) %画小波包树结构的图形
rcfs＝wprcoef(t，d，[2，1])；%重构小波包的结点(2，1)
figure(1)；
```

```
subplot(212); plot(rcfs);
title('重构的小波包结点(2,1)');
```

输出结果(如图 2.58、2.59 所示)：

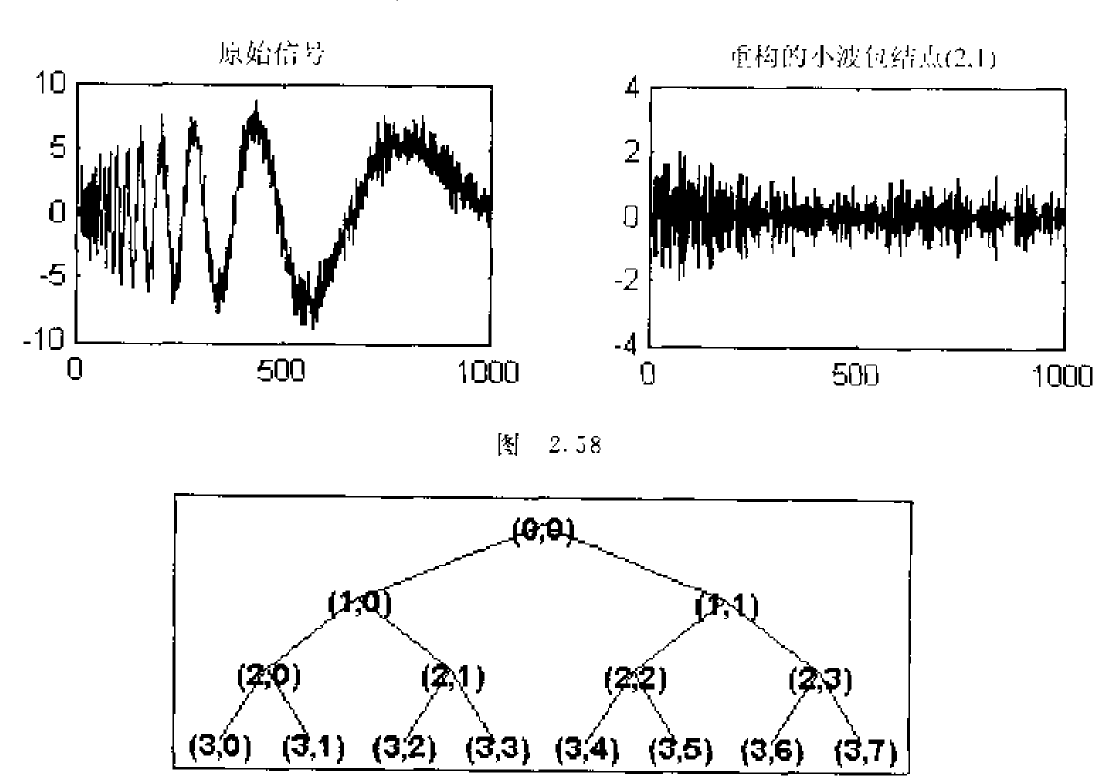


图 2.58

图 2.59

参见：maketree,wpdec,wpdec2,wprec,wprec2

7. wpfun

功能：小波包函数

格式：① [WPWS,X]=wpfun('wname',NUM,PREC)

　　　② [WPWS,X]=wpfun('wname',NUM)

说明：wpfun 是一个小波包分析函数。格式①计算指定小波'wname'的小波包，且时间长度为二进制时间间隔(长度为 2^{prec}，prec 必须是正整数)。输出矩阵 WPWS 包含从 0 到 NUM 的小波包函数，且按[W_0; W_1; …, W_{NUM}]的顺序排列，输出向量 X 是普通的 X -网格向量。[WPWS,X]=wpfun('wname',NUM)和[WPWS,X]=wpfun('wname',NUM,7) 是等价的。

当用正交小波时，产生小波包的方法是很简单的。首先，我们根据相应的小波，用两个长为 2N 的滤波器(h(n)和 g(n))开始，它们分别是低通分解滤波器和高通分解滤波器各自除以 $\sqrt{2}$ 后的重构滤波器。

我们先定义下面的函数序列($W_n(x)$，$n=0,1,2,\cdots$)

$$W_{2n} = 2\sum_{k=0}^{2N-1} h(k)W_n(2x-k)$$

$$W_{2n+1} = 2\sum_{k=0}^{2N-1} g(k)W_n(2x-k)$$

其中，$W_0(x)=\phi(x)$是尺度函数，$W_1(x)=\Psi(x)$是小波函数。

例如，对 haar 小波函数，当 $N=1$ 时，

$$h(0) = h(1) = 1/2,\ g(0) = -g(1) = 1/2$$

等式变为

$$W_{2n}(x) = W_n(2x) + W_n(2x-1)$$

$$W_{2n-1}(x) = W_n(2x) - W_n(2x-1)$$

在这里，$W_0(x)=\phi(x)$是 Haar 尺度函数，$W_1(x)=\Psi(x)$是 Haar 小波函数，两个函数的支撑长度均在区间[0,1]上。从上式我们可以看出，可以通过把支撑区间分别在[0,1/2]和[1/2,1]内的两个 1/2 尺度的 W_n 加起来获得 W_{2n}函数。同样，可以通过把支撑区间分别在[0,1/2]和[1/2,1]内的两个 1/2 尺度的 W_n 相减获得 W_{2n+1}函数。

对于更规则的小波，用相似的构造方法，可以获得光滑的小波包函数序列，且具有的支撑为[0，2N−1]。

举例：%计算 db2 的小波包函数 W_n，$n=0$，1，…，7

```
[wp,x]=wpfun('db2',7);
for i=1:8
    w=wp(i,:);
    subplot(4,2,i); plot(w);
    Ylabel(['W',num2str(i)]);
end
```

输出结果(如图 2.60 所示)。

参见：wavefun，waveinfo

8. wpsplt

功能：分割(分解)小波包

格式：① [T,D]=wpsplt(T,D,N)

② [T,D,CA,CD]=wpsplt(T,D,N)

③ [T,D,CA,CH,CV,CD]=wpjoin(T,D,N)

说明：wpsplt 是一个一维或二维小波包分析函数。它在重组一个结点后，便更新树结构和数据结构。格式①根据指定的重组结点 N，修改计算树结构 T 和数据结构 D；格式②除了返回格式①中的参数外，另外还返回结点的系数，其中，CA 是结点 N 的低频系数，CD 是结点 N 的高频系数；格式③除了返回格式①中的参数外，另外还返回结点的系数，其中，CA 是结点 N 的低频系数，CH、CV 和 CD 是结点 N 的高频系数。

举例：%装入信号

```
load noisdopp; x=noisdopp(1:1000);
%用 db1 小波包对信号 x 进行三层分解
[t,d]=wpdec(x,3,'db1','shannon');
plottree(t) %画小波包树结构的图形
```

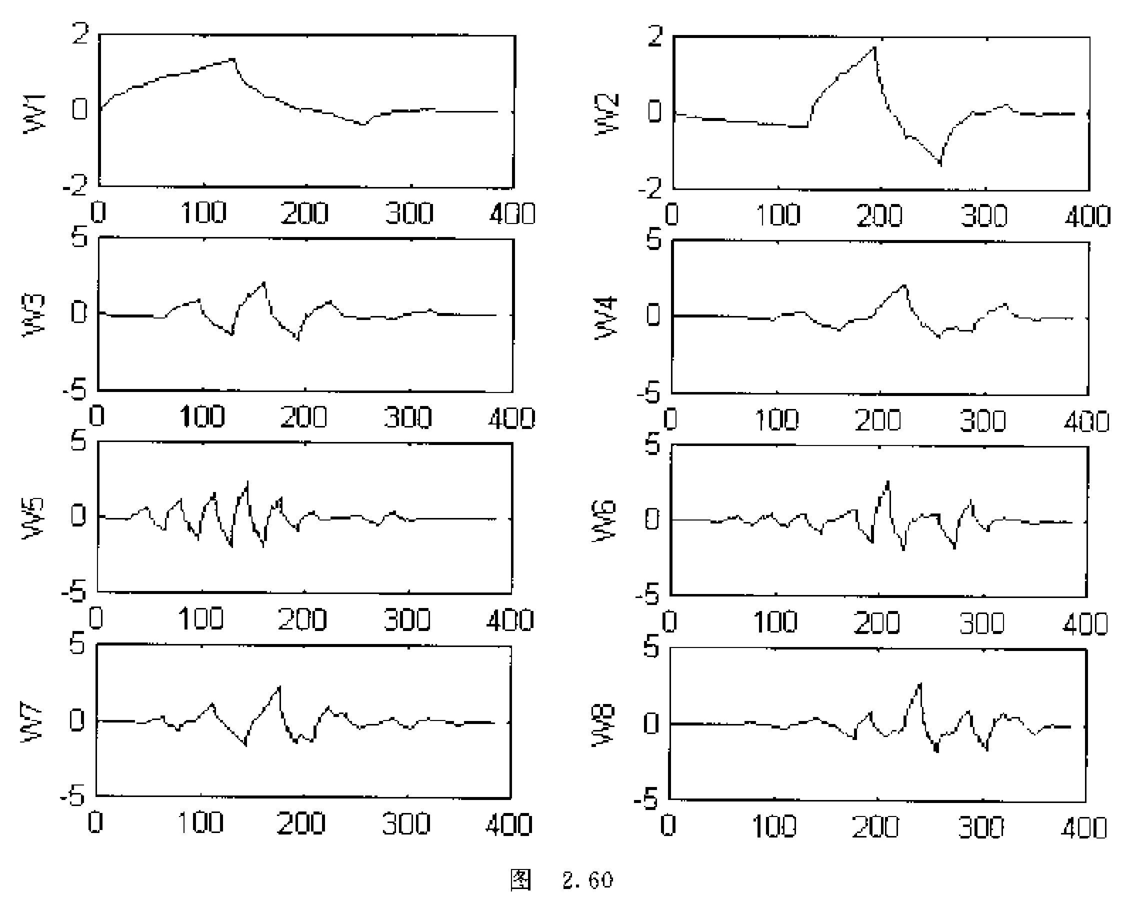


图　2.60

```
%重新分解小波包结点(3,0)或第 7 个结点
[wpt,wpd]=wpsplt(t,d,[3,0]);
%或等价于[wpt,wpd]=wpsplt(t,d,7);
plottree(wpt) %画小波包树结构的图形
```

输出结果(如图 2.61 所示)：

参见：maketree，wavedec，wavedec2，wpdec，wpdec2，wpjoin

9. wpjoin

功能：重新组合小波包

格式：① [T,D]=wpjoin(T,D,N)

② [T,D,X]=wpjoin(T,D,N)

③ [T,D]=wpjoin(T,D)

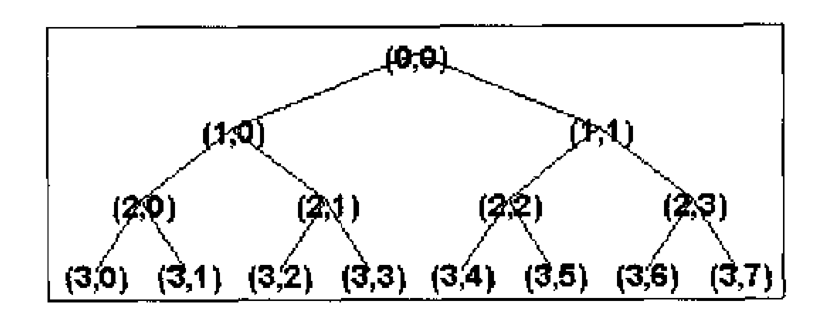
④ [T,D,X]=wpjoin(T,D)

说明：wpjoin 是一个一维或二维小波包分析函数，它用来重新组合小波包，即把结点 N(结点可以以索引的形式表示，也可以用深度一位置的形式表示)以下的二叉子树去掉(更新后的小波包分解树不对结点 N 作进一步的分解)后，返回一个更新后的小波包分解结构。

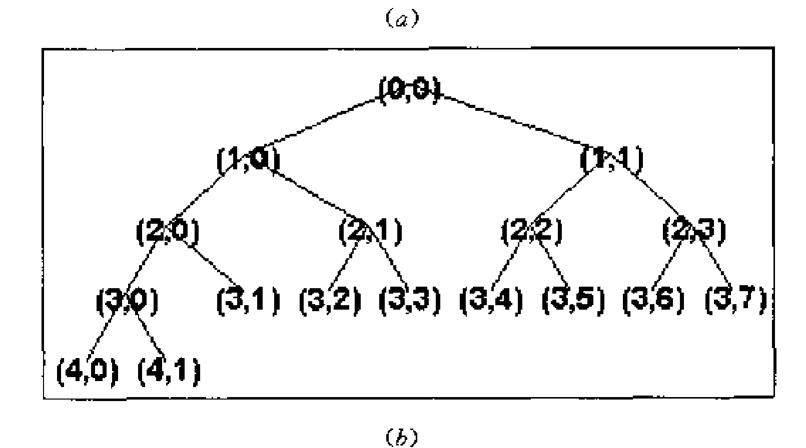
格式①根据指定的结点 N，去掉结点 N 以下的二叉子树后，返回修改计算后的小波包分解树结构 T 和数据结构 D(参见 maketree)。

格式②除了返回格式①中的参数外，另外还返回结点的小波包分解系数。

格式③等价于[T,D]=wpjoin(T,D,0)，它去掉了根结点以下的所有二叉子树，更新



(a)



(b)

图 2.61

后的小波包分解结构实质上对信号没有进行任何尺度的小波包分解。

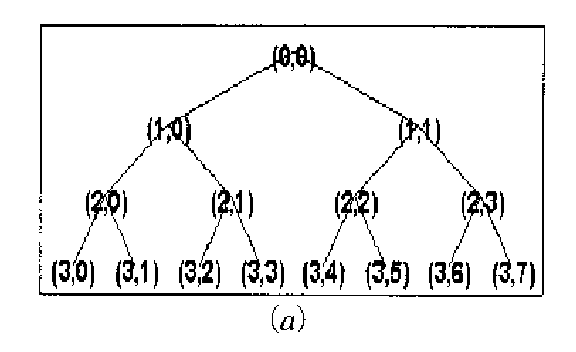
格式④等价于[T,D,X]=wpjoin(T,D,0)，更新后的小波包分解结构实质上对信号没有进行任何尺度的小波包分解，返回后的小波包分解系数 X 同原始信号一样。

举例：%装入信号

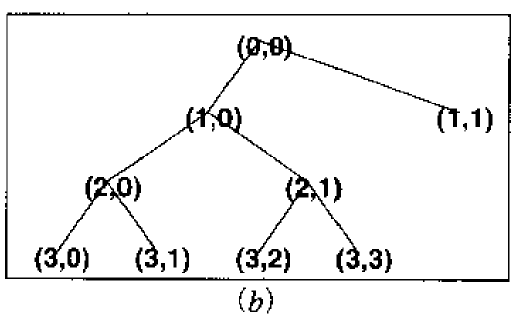
```
load noisdopp; x=noisdopp;
figure(1);
subplot(321); plot(x);
title('原始信号');
%用 db1 小波包对信号 x 进行三层分解
[wpt,wpd]=wpdec(x,3,'db1');
plottree(wpt) %画小波包树结构的图形
%重组小波包(1,1)或第 2 个结点，并返回更新后的小波包分解结构
%参数 c 是结点 2 的小波包分解系数
[wpt,wpd,c]=wpjoin(wpt,wpd,2); %2 是指结点 2，也可以用[1,1]表示
plottree(wpt) %画小波包树结构的图形
%以图形的方式显示分解系数的大小
figure(1);
subplot(322); plot(c);
title('结点 2 的小波包分解系数');
```

输出结果(如图 2.62、2.63 所示)。

参见：maketree,wpdec,wpdec2,wpsplt



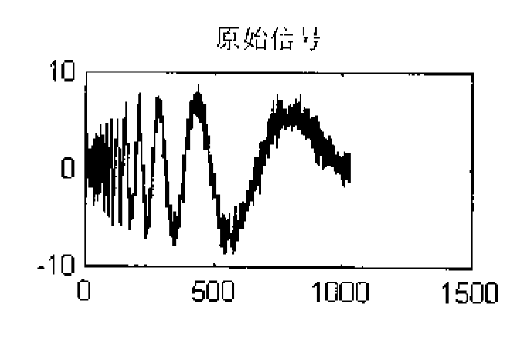
(a)



(b)

图　2.62

(a) 完整小波包分解树结构；(b)更新后的小波包分解树结构



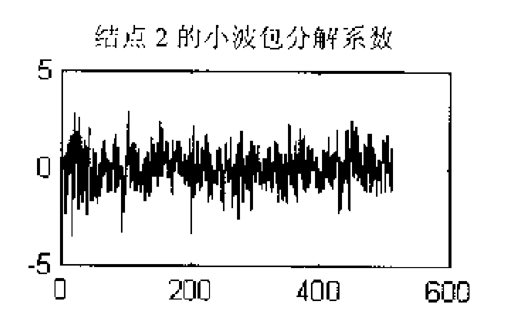


图　2.63

10. wpcutree

功能：剪切小波包分解树

格式：① [T,D]=wpcutree(T,D,L)

　　② [T,D,RN]=wpcutree(T,D,L)

说明：wpcutree 是一个一维或二维的小波包分析函数，它将小波包分解树第 L 层以下的所有二叉子树全部剪切掉，并返回经过修改计算后的小波包分解结构[T,D]，其中，L 是小波包分解树的层数。

对于格式①，它将剪切掉小波包分解第 L 层以下所有的二叉子树，并返回经过修改计算后的相应小波包分解结构[T,D]。

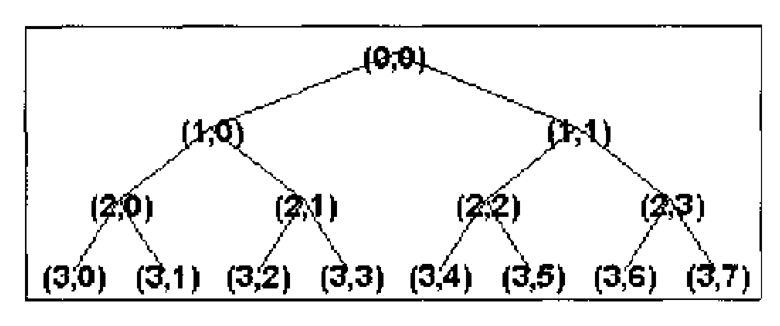
对于格式②，它除了返回格式①中变量外，还返回向量 RN。向量 RN 为初始小波包分解树，经过重组的树结点索引序号。

举例：%装入信号

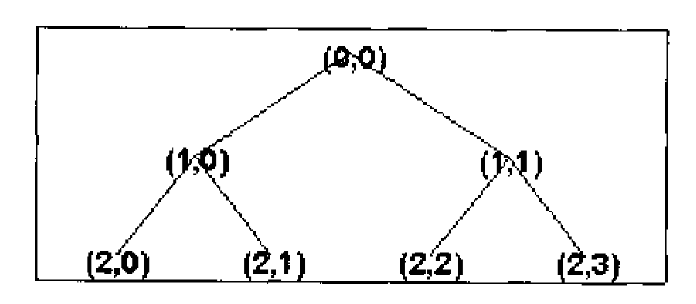
```
load noisdopp; x=noisdopp;
%用 db1 小波包分解信号 x 到第三层
[wpt,wpd]=wpdec(x,3,'db1');
plottree(wpt) %画小波包树结构的图形
%在第二层剪切小波包树
[nwpt,nwpd,rn]=wpcutree(wpt,wpd,2); %2 是指第二层
plottree(nwpt) %画新的小波包树结构(nwpt)的图形
```

```
disp('初始树中经过重组结点的索引序号：')
rn
```

输出结果(如图 2.64 所示)：



(a) 初始完整小波包分解树



(b) 经过剪切后小波包分解树

图　2.64

```
初始树中经过重组结点的索引序号：
rn =
     3
     4
     5
     6
```

参见：maketree，wpdec，wpdec2

11. besttree

功能：计算最佳(优)树

格式：① [TT，DD]=besttree(T，D)

② [TT，DD，E]=besttree(T，D)

③ [TT，DD，E，N]=besttree(T，D)

说明：besttree 是一个一维或二维小波包分析函数。它能根据一个熵标准来计算初始树的最佳子树，算出的子树比初始树小得多。

根据小波包的组织方式，对于一个给定的正交小波，一个长度为 $N=2^{J}$ 的信号最多可以有 2^{L} 种不同的分解方式，这恰好是一个深度为 L 的完整的二叉子树的数目，是一个非常庞大的数目，一般说来，用枚举法进行一一列举是难以想象的。由于我们感兴趣的是根据一个简单而又可行的标准来寻找一个最佳的分解方式(即最佳分解树结构或最佳小波包基)

和一个有效的算法，所以我们可以根据最小熵标准来进行处理。

对于格式①，它是根据小波包分解结构[T,D]来计算小波包分解的最佳树 TT 及相应的数据结构 DD。

对于格式②，它返回计算后的最佳树结构 TT、数据结构 DD 以及初始树每个结点的熵值向量 E(向量元素的顺序与结点的索引序号依次对应，即索引为 0、1、2 …的结点熵值，依次对应向量中第 1、2、3 …个元素的值)。

对于格式③，它除了返回格式②中所返回的参数外，同时还返回最佳树与初始树相比，所有被合并结点的索引序号的向量 N。例如，若返回向量 N=[2，6]，则表示初始树中索引序号为 2 和 6 的结点以下的二叉子树被合并，即最佳树与初始树相比，索引为 2 和 6 的结点不再分解。

下面我们对算法进行一个简单的说明：考虑一维情况时，首先从根结点开始，用以下方法来计算最优树，即当 N1 和 N2 的熵之和小于 N 的熵时，一个结点 N 可以被分成两个结点 N1 和 N2。当然，这仅仅是基于结点 N 的可得信息的一种局部标准。有几种熵标准可供选择(详见 wentropy 函数)。如果 wentropy 函数沿着小波包系数的方向是一个递增的函数，那么这种算法就可以得到最优树。

首先从初始树 T 开始，用这种合并边界结点的算法，就可以从 T 中所有的二叉子树中得到最佳树。

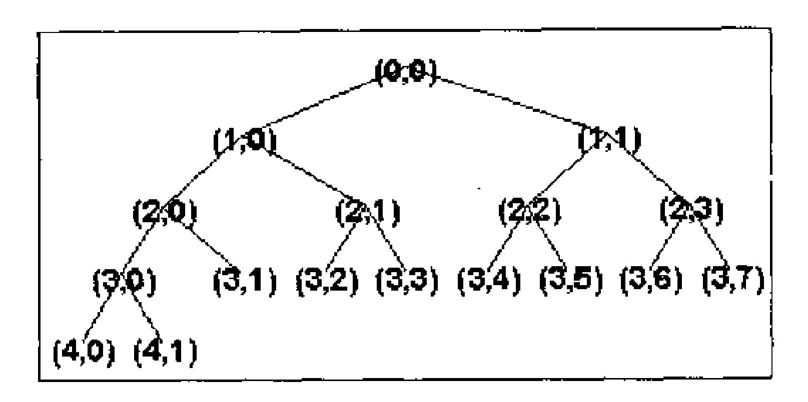
举例：%装入信号

```
load noisdopp; x=noisdopp;
%用 db1 小波包将信号 x 进行三层分解
[wpt,wpd]=wpdec(x,3,'db1');
%用默认的(shannon)熵分解小波包[3,0]
[wpt,wpd]=wpsplt(wpt,wpd,[3,0]);
%画出小波树的结构 wpt
plottree(wpt)
%计算最佳树
[bt,bd,e,n]=besttree(wpt,wpd);
%画出最佳树的结构 bt
plottree(bt)
disp('表示初始树的各个结点的熵值向量为：');
e
disp('初始树中被合并的结点索引序号向量 n 为：');
n
```

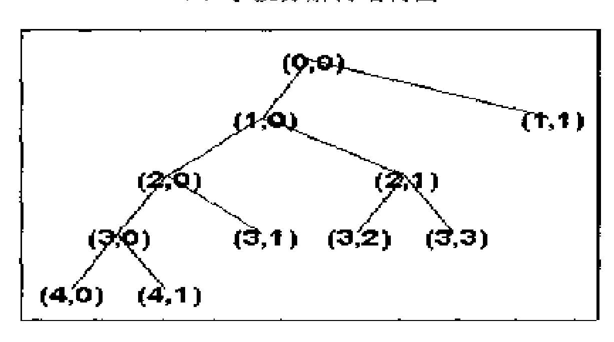
输出结果(如图 2.65 所示)。

表示初始树的各个结点的熵值向量为：

```
e =
  1.0e+004 *
  Columns 1 through 7
    -9.8173    -9.7822    -0.0350    -9.7303    -0.0519    -0.0205
```



(*a*) 小波分解树结构图



(*b*) 最佳树结构

图　2.65

```
-0.0142
   Columns 8 through 14
     -9.5880    -0.1423    -0.0318    -0.0200    -0.,0109    -0.0096
-0.0053
   Columns 15 through 21
     -0.0089    -9.3112    -0.2768        NaN        NaN        NaN
   NaN
   Columns 22 through 28
         NaN        NaN        NaN        NaN        NaN        NaN
   NaN
   Columns 29 through 31
         NaN        NaN        NaN
```

初始树中被合并的结点索引序号向量 n 为

```
n =
     2
```

从图 2.65(*a*)和图 2.65(*b*)中可以看出，计算出的最佳子树比初始树要小得多。由于初始树中，只有索引为 2 的结点以下的二叉子树被合并，所以 n 为一个单元素向量。

参见：bestlevt，maketree，wentropy，wpdec，wpdec2

12. bestlevt

功能：计算最佳完整小波包树

格式：① [TT,DD]=bestlevt(T,D)

② [TT,DD,E]=bestlevt(T,D)

说明：bestlevt 函数是一个一维或二维的小波包分析函数。它可以根据一种熵标准计算出初始树的最佳完整子树，这个完整的子树比初始树的深度小一些。格式①根据最优深度小波包树的分解，修改计算小波包分解结构[T,D]，并返回新的小波包分解结构[TT,DD]；格式②除了返回格式①的参数外，它还返回最优熵值 E。

bestlevt 函数在功能上与 besttree 函数类似，它们之间的不同之处是：bestlevt 函数返回的树结构 TT 首先是一个完整的二叉子树，然后在完整的基础上，它是一个最佳小波包分解树。besttree 函数返回的是一个从熵标准角度出发，最佳的小波分解树，但它不一定是一个完整的二叉子树；bestlevt 函数返回的最佳熵值 E 是最佳分解树的总体熵，是一个数值。besttree 函数返回的是初始树每一个结点的熵值，是一个向量。

举例：%装入信号

```
load noisdopp; x=noisdopp;
%用 db1 小波包对信号 x 进行三层分解
[wpt,wpd]=wpdec(x, 3, 'db1');
%采用默认的(shannon)熵，分解小波包[3,0]
[wpt,wpd]=wpsplt(wpt,wpd,[3,0]);
plottree(wpt) %画出小波包的树结构 wpt 的图形
[blt,bld]=besttree(wpt,wpd); %计算最佳层次的树
plottree(blt) %画出最佳层次的树结构 blt 的图形
```

输出结果(如图 2.66 所示)。

参见：besttree,maketree,wentropy,wpdec,wpdec2

13. wp2wtree

功能：从小波包树中提取小波树

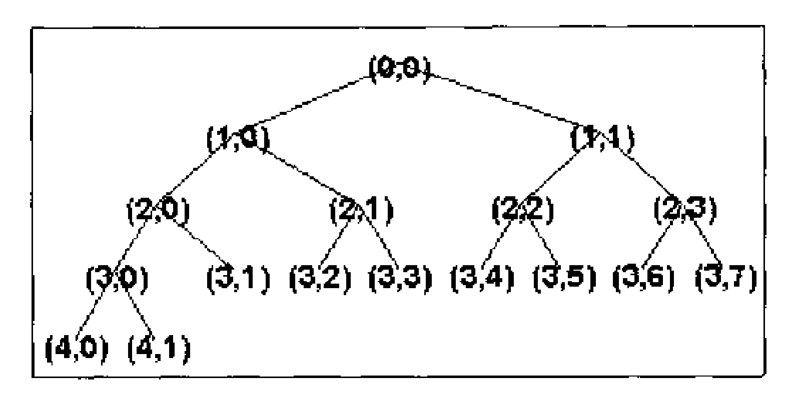
格式：[T,D]=wp2wtree(T,D)

说明：wp2wtree 是一个一维或二维的小波包分析函数，它根据小波包分解树，修改计算树结构 T 和数据结构 D，并返回小波树[T, D]。

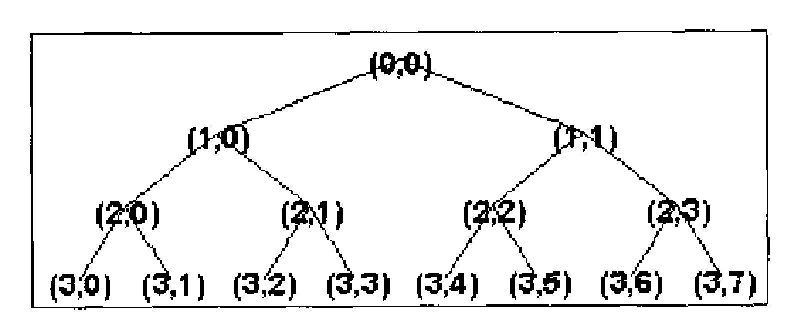
举例：%装入信号

```
load noisdopp; x=noisdopp;
%用 db1 小波包将信号 x 分解到第三层
[wpt,wpd]=wpdec(x,3,'db1');
plottree(wpt) %画小波包树的图形
[wt,wd]=wp2wtree(wpt,wpd); %计算小波树
plottree(wt) %画小波树的图形
```

输出结果(如图 2.67 所示)。

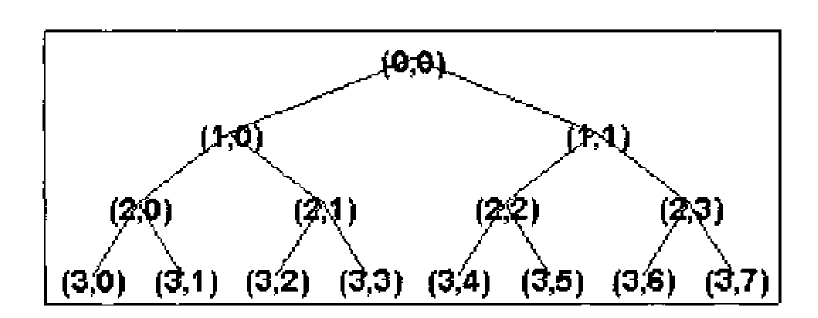


(a)

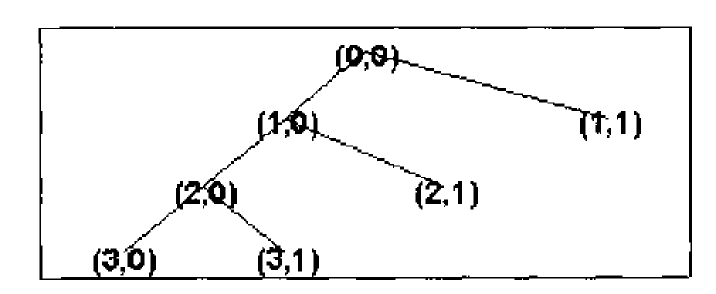


(b)

图 2.66



(a)



(b)

图 2.67

参见：maketree，wpdec，wpdec2

14. wentropy

功能：计算小波包的熵

格式：① E=wentropy(X,T,P)

　　② E=wentropy(X,T)

说明：wenttopy 是一个一维和二维小波包分析函数。

对于格式①，它返回输入向量或矩阵 X 的熵值 E。在 X 为向量或矩阵这两种情况下，输出 E 都是一个实数。参数 T 用来指定一个熵标准，它有'Shannon'、'threshold'、'norm'、'log energy'、'sure'和'user'这几种类型，P 是一个可选的参数，它的选择根据参数 E 的值来决定。

- 如果 T='shannon'或'log energy'时，则 P 不用；
- 如果 T='threshold' 或'sure'时，则 P 是阈值，并且必须为正数；
- 如果 T='norm'时，则 P 是指数，且有 1≤P≤2；
- 如果 T='user'时，则 P 是一个包含 *.M 文件名的字符串，这个 *.M 文件是在一个输入变量 X 下，用户自己的熵函数。

对于格式②，它等价于 E=wentropy(X,T,0) 。

可以证明，可加性非常有利于进行二叉树的高效搜索和它的基本分割。传统的基于熵的标准可以完成这项工作，它可以对给定的信号进行信息相关的性能描述。熵的概念在许多领域中是一个普遍的概念，特别是在信号处理中。下面让我们列举常用的四个熵标准，另外，许多其它的熵标准也可以很容易地组合起来(参见 wentropy)。在下面的叙述中，用 s 代表信号，用 s_i 代表信号 s 在一个正交小波包基上的投影系数。熵 E 必须是一个递增的价值函数，即 $E(0)=0$，$E(s)=\sum_i E(s_i)$ 。

(1) Shannon 熵：

$\mathrm{E1}(s_i) = -s_i^2\log s_i^2$　所以有

$\mathrm{E1}(s) = -\sum_i s_i^2\log s_i^2$　约定 $0\log 0=0$。

(2) l^P范数($1\leqslant P<2$)：

$E2(s_i) = |s_i|^P$　所以有

$E2(s) = \sum_i |s_i|^P = \| s \|_P^P$

(3) “对数能量”(log energy)熵：

$E3(s_i) = \log s_i^2$　所以有

$E3(s) = \sum_i \log s_i^2$　约定 $\log 0=0$。

(4) 阈值熵：

如果$|s_i|>\varepsilon$，则 $E4(s_i)=1$；其余，$E4(s_i)=0$。所以有

$E4(s)$为信号大于阈值 ε 的时间点的个数。

(5) “SURE”熵：

$E5(s) = n - \# + \sum_i \min(s_i^2、p^2)$

(如果$|s_i|\leqslant p$，则#=i)

其中，如果$|s_i|\leqslant p$时，则#=i。

举例：x=rand(1,200)；%产生一个初始的随机信号

```
%计算信号x的shannon熵
e1=wentropy(x,'shannon')
%计算信号x的对数能量熵
e2=wentropy(x,'log energy')
%计算信号x的阈值熵，阈值等于0.2
e3=wentropy(x,'threshold',0.2)
%计算信号x的SURE熵，确定阈值等于3
e4=wentropy(x,'sure',3)
%计算信号x的范数熵，范数指数等于1.1
e5=wentropy(x,'norm',1.1)
%计算信号x的用户熵，熵函数由用户自定义，
%例如userent函数，该函数必须是一个M文件，并且第一行是下列形式
%function e=userent(x) %式中x是一个向量，e是一个实数
%这样，新的熵函数就定义了，可以用下列形式调用：
%e6=wentropy(x,'user','userent')
```

输出结果：

```
e1 =
    48.8337
e2 =
  -353.2800
e3 =
    168
e4 =
  -131.0999
e5 =
    99.6449
```

15. entrupd

功能：更新小波包的熵值

格式：① NDATA=entrupd(TREE,DATA,T)

② NDATA=entrupd(TREE,DATA,T,P)

说明：entrupd函数是一个一维或二维小波包应用函数。

对于格式①，根据一个给定的小波包分解结构[TREE,DATA](参见maketree函数)和熵标准T(参见wentropy函数)，返回更新后的小波包分解数据结构NDATA。此时，各结点的熵值发生了变化，可以用wdatamgr函数读取各结点的熵值大小。

将格式②和格式①相比，它多了一个可选择输入参数P(其用法与wentropy函数中的

P 参数相同)。它返回的也是一个更新后的小波包分解数据结构 NDATA，同样可以用 wdatamgr 函数读取各结点的熵值大小。

举例：%装入信号

```
load noisdopp; x=noisdopp;
%用 db2 小波包对信号 x 进行二层分解，用 shannon 熵作为熵标准
[t,d]=wpdec(x,2,'db1','shannon');
%计算小波包分解树的结点
nodes=allnodes(t);
%读取初始小波包分解数据结构中各结点的熵值
disp('初始树各结点的熵值:');
ent=wdatamgr('read_ent',d,nodes)
%不改变树结构和数据结构，选用'threshold'熵标准(阈值为 0.5)，更新结点的熵值
d=entrupd(t,d,'threshold',0.5);
%读取更新后小波包分解数据结构中各结点的熵值
disp('更新后各结点的熵值:');
nent=wdatamgr('read_ent',d,nodes)
```

输出结果：

```
初始树各结点的熵值：
ent =
  1.0e+004 *
  -5.8615    -6.8204    -0.0350    -7.7901    -0.0497    -0.0205
-0.0138
更新后各结点的熵值：
nent=
     937   488   320   241   175   170   163
```

参见：wentropy，wpdec，wpdec2

2.6　信号和图像的消噪与压缩

在这一节里，我们将详细讲述关于小波或小波包在信号和图像的消噪与压缩方面应用的一些函数，通过对这些函数的学习，我们基本上可以掌握如何对信号或图像进行消噪与压缩处理。

1. ddencmp

功能：获取在消噪或压缩过程中的默认值阈值(软或硬)、熵标准

格式：① [THR,SORH,KEEPAPP,CRIT]=ddencmp(IN1,IN2,X)

② [THR,SORH,KEEPAPP]=ddencmp(IN1,'wv',X)

③ [THR,SORH,KEEPAPP,CRIT]=ddencmp(IN1,'wp',X)

说明：ddencmp 是一个获取在消噪或压缩过程中的默认值阈值的函数，即在运用小波或小

波包进行一维或二维信号的消噪或压缩时，它可以给出默认阈值。X 为一维(向量)或二维(矩阵)信号；IN1 是消噪与压缩的选择，'den'表示消噪，'cmp'表示压缩；IN2 是小波与小波包的选择，'wv'表示小波，'wp'表示小波包；THR 为返回的阈值；SORH 为软阈值和硬阈值选择参数；KEEPAPP 让你保存低频信号；CRIT(只在小波包分析时用)是熵名(即熵标准的选择，参见 wentropy)。

格式①为一般格式，它根据输入的向量或矩阵 X(对应一维或二维信号)，返回的是消噪或压缩的默认值。

格式②为小波分析(有三个输出变量)的情况，如果 IN1='den'，则返回信号 X 的消噪默认阈值；如果 IN1='cmp'，则返回信号 X 的压缩默认阈值。这些值可以用于函数 wdencmp。

格式③为小波包分析(有四个输出变量)的情况，如果 IN1='den'，则返回信号 X 的消噪默认阈值；如果 IN1='cmp'，则返回信号 X 的压缩默认阈值。这些值可以用于函数 wpdencmp。

举例：

```
%产生高斯白噪声
init=2055415866; randn('seed',init);
x=randn(1,1000);
%求取小波分析的默认值(三个输出变量)，
%这些值可以应用于 wdencmp(带参数'gbl')
%求取信号消噪的默认阈值、软阈值，并且保留低频系数，
%thr=sqrt(2*log(n))*s，其中 s 是噪声层的估计值
[thr1,sorh1,keepapp1]=ddencmp('den','wv',x)
%求取信号压缩的默认阈值、硬阈值，并且保留低频系数，
%thr=median(abs(第一层的高频系数))，如果它不为 0，则
%thr=0.05*max(abs(第一层的高频系数))
[thr2,sorh2,keepapp2]=ddencmp('cmp','wv',x)
%求取小波包分析的默认值(四个输出变量)，这些值可以应用于 wdencmp
%求取信号消噪的默认阈值、软阈值，并且保留低频系数，
%thr=sqrt(2*log(n*log(n)/log(2)))，其中噪声层被认为等于 1
%默认的熵标准是'sure'标准
[thr3,sorh3,keepapp3,crit3]=ddencmp('den','wp',x)
%求取信号压缩的默认阈值、硬阈值，并且保留低频系数，
%thr=median(abs(第一层的高频系数))
%默认的熵是'threshold'标准
[thr4,sorh4,keepapp4,crit4]=ddencmp('cmp','wp',x)
```

输出结果：

```
thr1 =
    3.8593
sorh1 =
s
```

```
keepapp1 =
      1
thr2 =
     0.7003
sorh2 =
h
keepapp2 =
      1
thr3 =
     4.2911
sorh3 =
h
keepapp3 =
      1
crit3 =
sure
thr4 =
     0.7003
sorh4 =
h
keepapp4 =
      1
crit4 =
threshold
```

参见：wdencmp，wentropy，wpdencmp

2. thselect

功能：信号消噪的阈值选择

格式：THR=thselect(X,TPTR)

说明：thselect 是一个一维消噪函数，它根据信号 X 和一个阈值选择标准，来确定一个消噪的阈值。该函数返回的是对 X 进行消噪处理中所采用的自适应阈值，阈值的选取规则由字符串 TPTR 定义，TPTR 有如下四种规则(见表 2-4)。

阈值选择规则是基于基本模型 y=f(t)+e，其中 e 是白噪声 N(0,1)，对于方差未知的噪声或非白噪声可以重新调节输出阈值 THR(参见 wden 中的 SCAL 参数)。

可以选择以下几种阈值：

• TPTR='rigrsure'是一种基于史坦的无偏似然估计(二次方程)原理的自适应阈值选择。对一个给定的阈值 t，得到它的似然估计，再将非似然 t 最小化，就得到了所选的阈值，它是一种软件阈值估计器。

表 2-4 阈值选择规则

TPTR 的选项	阈值选择规则
'rigrsure'	采用史坦(Stein)的无偏似然估计(Unbiased Risk Estimate)原理(SURE)进行自适应阈值选择
'sqtwolog'	固定的阈值形式，它等于 sqrt(2 * log(length(s)))
'heursure'	启发式阈值选择
'minimaxi'	用极大极小原理选择的阈值

- TPTR='sqtwolog'采用的是固定的阈值形式，产生的阈值大小是 sqrt (2 * log (length(X)))。
- TPTR='heursure'是启发式阈值选择，是最优预测变量阈值选择。如果信噪比很小，SURE 估计有很大的噪声，在这种情况下，就采用这种固定的阈值。
- TPTR='minimaxi'采用的是极大极小原理选择阈值，它产生一个最小均方误差的极值，而不是无误差。在统计学上，这种极值原理用于设计估计器。因为已经被消噪的信号可以看作与未知回归函数的估计式相似，这种极值估计器可以在一个给定函数集中实现最大均方误差最小化。

如果信号 y 表示的是一个高斯白噪声信号 N(0,1)，我们来看在各个阈值选择规则下 thselect 命令函数是如何选取阈值的。

举例：%产生高斯白噪声

```
init=2055415866; rand('seed',init);
x=randn(1,1000);
%对每一个阈值选取规则，求取阈值
%采用 SURE 的自适应阈值
thr1=thselect(x,'rigrsure')
%固定形式的阈值
thr2= thselect(x,'sqtwolog')
%启发式阈值
thr3= thselect(x,'heursure')
%极大极小原理阈值
thr4= thselect(x,'minimaxi')
```

输出结果：

```
thr1 =
    2.7316
thr2 =
    3.7169
thr3 =
    3.7169
thr4 =
```

2.2163

参见：wden

3. wden

功能：用小波进行一维信号的自动消噪

格式：① [XD,CXD,LXD]=wden(X,TPTR,SORH,SCAL,N,'wname')

② [XD,CXD,LXD]=wden(C,L,TPTR,SORH,SCAL,N,'wname')

说明：wden 是一个利用小波对一维信号进行自动消噪的函数。其中，X 为原始信号，TPTR 为阈值选择规则。

- TPTR='rigrsure'时，采用史坦(Stein)的无偏似然估计原理。
- TPTR='heursure'时，是启发式阈值选择。
- TPTR='sqtwolog'时，是固定阈值 sqrt(2 * log(·))。
- TPTR='minmaxi'时，用于极大极小值原理进行阈值的选择(参见 thselect)。

SORH 是软阈值或硬阈值的选择(参见 wthresh)。

- SORH='s'时为软阈值。
- SORH='h'时为硬阈值。

SCAL 定义所乘的阈值是否要重新调整：

- SCAL='one'时，不用重新调整。
- SCAL='sln'时，根据第一层的系数进行一次噪声层的估计来调整阈值。
- SCAL='mln'时，在不同层估计噪声层，以此来调整阈值。

N 为小波分解的层数。'wname'是一个指定的正交小波名(参见 wmaxlev 和 wfilters)。XD 为消噪后的信号，[CXD，LXD]为消噪后信号 XD 的小波分解结构。

格式①返回经过对输入信号 X 的小波分解系数(N 层分解)进行阈值门限处理后的消噪信号 XD，同时还返回消噪信号 XD 的小波分解结构[CXD,LXD](参见 wavedec)。格式②返回参数同格式①，不同的是它是直接对含噪声信号的小波分解结构[C,L]进行阈值处理而得到的。

一个含噪声信号的基本模型是下面的形式：

$$s(n)=f(n)+\sigma e(n)$$

式中，时间 n 是等时间间隔的。

在最简单的模型中，认为 e(n)是高斯白噪声 N(0,1)，并且噪声水平 σ 认为等于 1。消噪的目的是减小噪声部分的值，以恢复信号 f。

噪声的消除可按以下三个步骤：

1　选择小波和小波分解的层次，计算信号 s 到第 N 层的小波分解。

2　高频系数的阈值选择。对于从第 1 到第 N 层的每一层，选择一个阈值，并且对高频系数用软阈值进行处理。

3　根据第 N 层的低频系数和从第一层到第 N 层的经过修改的高频系数，计算出信号的小波重建。

对于阈值的选择，可以参见 thselect，需要指出的是：

- 分解高频系数的向量是信号 f 的系数和噪声 e 的系数的叠加，对噪声 e 的分解导致

高频系数是一个标准的高斯白噪声。

• 当信号f的高频部分在噪声域很小时，阈值规则Minimaxi和SURE更加保守(即不容易丢失信号中的有用成份，但只除去较少的噪声)和更方便一些。另外两种阈值选择规则可以更有效地去除噪声。而'heursure'是一种折衷的办法。

举例：

```
snr=3 %设置信噪比
init=2055615866 %设置随机数的初始值
%产生一个 Heavy sine 初始信号 x 和含标准的高斯白噪声的信号 xref
[xref,x]=wnoise(3,11,snr,init);
%将信号 x 用 sym8 小波分解到第 5 层，并对高频系数用 heursure 软阈值
%对分解系数进行阈值处理，以消除噪声信号
lev=5;
xd=wden(x,'heursure','s','one',lev,'sym8');
%画信号图形
subplot(321); plot(xref);
axis([1 2048 -10 10]);
title('原始信号');
subplot(322); plot(x);
axis([1 2048 -10 10]);
title(['含噪声信号，信噪比为',num2str(fix(snr))]);
subplot(323); plot(xd);
axis([1 2048 -10 10]);
title('用 heursure 阈值去噪后的信号');
%用 rigrsure 阈值对噪声的标准偏差单层估计，来进行信号的去噪
xd=wden(x,'rigrsure','s','sln',lev,'sym8');
subplot(324); plot(xd);
axis([1 2048 -10 10]);
title(' 用 rigrsure 阈值去噪后的信号');
%用 sqtwolog 阈值对噪声的标准偏差单层估计，来进行信号的去噪
xd=wden(x,'sqtwolog','s','sln',lev,'sym8');
subplot(325); plot(xd);
axis([1 2048 -10 10]);
title(' 用 sqtwolog 阈值去噪后的信号');
%用 minimaxi 阈值对噪声的标准偏差多层估计，来进行信号的去噪
xd=wden(x,'minimaxi','s','sln',lev,'sym8');
subplot(326); plot(xd);
axis([1 2048 -10 10]);
title('用 minimaxi 阈值去噪后的信号');
%如果需要许多近似解，最好进行一次分解，多次阈值过滤
[c,l]=wavedec(x,lev,'sym8')      %分解
```

xd＝wden(c,l,'minimaxi','s','sln',lev,'sym8')；%对分解结构[c,l]进行过滤

输出结果(如图 2.68 所示)：

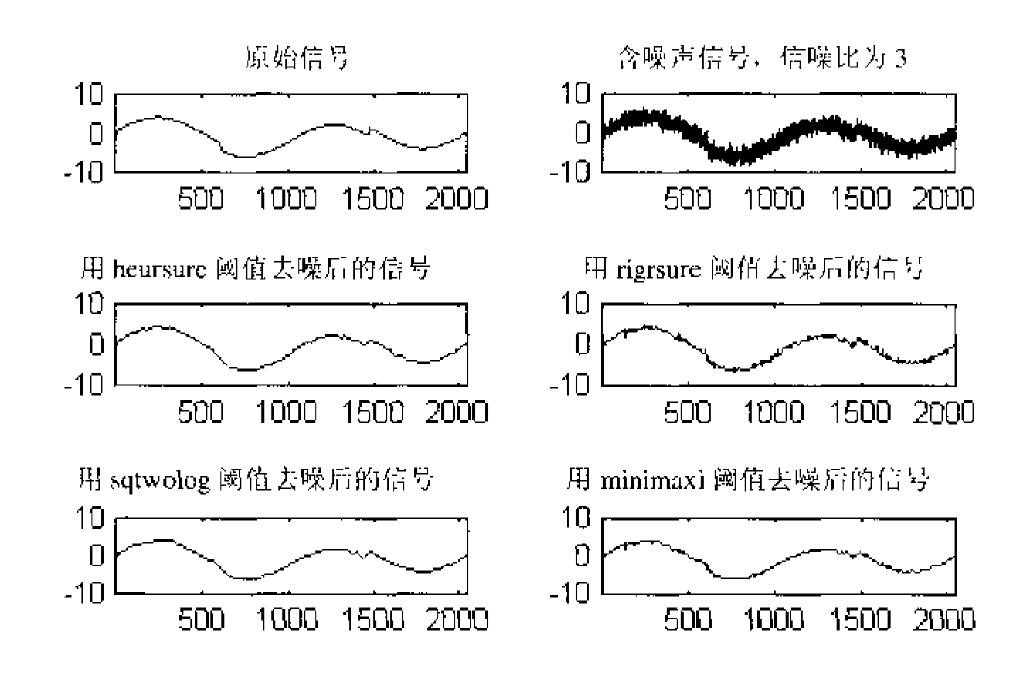


图　2.68

参见：thselect，wavedec，wdencmp，wfilters，wthresh

4. wdencmp

功能：用小波进行信号的消噪或压缩

格式：① [XC,CXC,LXC,PERF0,PERFL2]＝wdencmp('gbl',X,'wname',N，THR，SORH，EEPAPP)

② [XC，CXC，LXC，PERF0，PERFL2]＝wdencmp('lvd'，X，'wname'，N，THR，SORH)

③ [XC，CXC，LXC，PERF0，PERFL2]＝wdencmp('lvd'，C，L，'wname'，N，THR，SORH)

说明：wdencmp 是一个一维或二维消噪或压缩函数。它用小波对信号或图像进行消噪或压缩。格式①对输入信号 X(一维或二维)进行消噪或压缩后返回 XC(消噪或压缩后的结果)，其中，'wname'指定所用的小波函数，'gbl'(global 的缩写)表示各层都是用同一个阈值处理。另外，输出参数[CXC，LXC]是 XC 的小波分解结构(参见 wavedec 或 wavedec2)。PERF0 和 PERFL2 是恢复和压缩 L^2 范数百分比。如果[C，L]是 X 的小波分解结构，则 PERFL2 ＝ 100 * (CXC 向量的范数/C 向量的范数)^2；如果 X 是一个一维信号，小波'wname'是一个正交小波，则 $PERFL2=\frac{100\|XC\|^2}{\|X\|^2}$。N 表示小波分解的层数，'wname'是一个包含小波名的字符串(参见 wmaxlev 和 wfilters)，SORH('s'或'h')是软阈值或硬阈值的选择(参见 wthresh)。如果 KEEPAPP＝1，则低频系数不进行阈值量化(即系数不会受到改变)，反之，低频系数要进行阈值量化(即系数会受到改变)。

wdencmp('gbl',C,L,'wname',N,THR,SORH,KEEPAPP)具有同上面一样的输出参数和输入选择项，只不过它是直接从小波分解结构[C，L]中进行信号消噪或压缩处理。

• 对一维情况和'lvd' (level-dependent，即每层用一个不同的阈值)选项，如果用相同的输入选项，格式②和格式③具有相同的输出变量，但是每层必须都要有一个阈值，故阈值向量 THR 的长度为 N。另外，低频系数被保存。与 wden(自动消噪)相比，wdencmp 更灵活，可以按照你自己的消噪方案来消噪。

• 对二维情况和'lvd'选项，格式②和格式③中，THR 必须是一个三维矩阵，它含有水平、斜向、垂直三个方向的独立阈值，且长度为 N。

用一个给定的小波基进行压缩后，它意味着信号在小波域的表示相对缺少了一些信息。之所以能对信号进行压缩，是因为对于规则的信号，可以用很少的低频系数(在一个合适的小波层上)和一部分高频系数来近似表示。

与信号消噪类似，信号的压缩也分为以下几个步骤：

(1) 进行信号的小波分解。

(2) 将高频系数进行阈值量化处理。对从 1 到 N 的每一层高频系数，都可以选择不同的阈值，并且用硬阈值进行系数的量化。

(3) 对量化后的系数进行小波重构。

消噪和压缩的不同之处在第二步，具体参见第 3 章。

举例：

例 1： %装入电信号，选取一部分

```
load leleccum; indx=2600 : 3100; x=leleccum(indx);
%画出原始信号
subplot(221); plot(x);
title('原始信号');
%用固定阈值进行信号的压缩
thr=35;          [xd,cxd,lxd,perf0,perfl2]=wdencmp('gbl',x,'db3',2,thr,'h',1);
%画出压缩后的信号
subplot(222); plot(xd);
title('固定阈值压缩信号');
%用 wdencmp 进行电信号的消噪，采用默认值(参见 ddencmp)
[thr,sorh,keepapp]=ddencmp('den','wv',x)
%用全局阈值选项进行电信号的消噪
xd=wdencmp('gbl',x,'db3',2,tbr,sorb,keepapp);
%画出消噪后信号
subplot(223); plot(xd);
title('全局阈值消噪信号');
tbr=[24.345,27.212];
xd=wdencmp('lvd',x,'db3',2,tbr,'h');
subplot(224); plot(xd);
title('独立阈值消噪信号');
```

输出结果(如图 2.69 所示)：

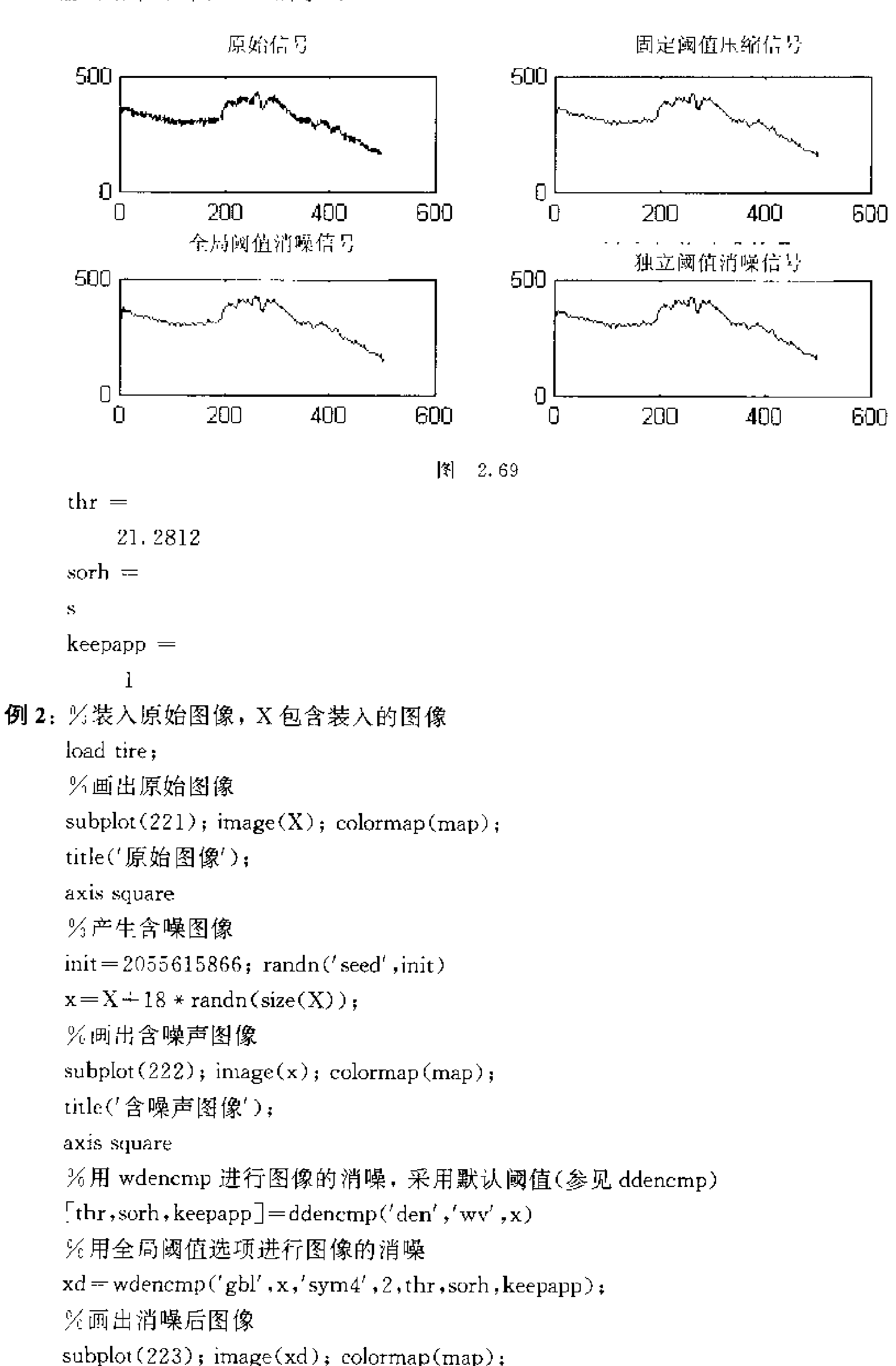


图　2.69

```
thr =
      21.2812
sorh =
s
keepapp =
       1
```

例 2：%装入原始图像，X 包含装入的图像

```
load tire;
%画出原始图像
subplot(221); image(X); colormap(map);
title('原始图像');
axis square
%产生含噪图像
init=2055615866; randn('seed',init)
x=X+18 * randn(size(X));
%画出含噪声图像
subplot(222); image(x); colormap(map);
title('含噪声图像');
axis square
%用 wdencmp 进行图像的消噪，采用默认阈值(参见 ddencmp)
[thr,sorh,keepapp]=ddencmp('den','wv',x)
%用全局阈值选项进行图像的消噪
xd=wdencmp('gbl',x,'sym4',2,thr,sorh,keepapp);
%画出消噪后图像
subplot(223); image(xd); colormap(map);
```

```
title('全局阈值消噪图像');
axis square
%用独立阈值选项进行图像的消噪
    thr_h=[96.245,97.411];     %水平方向阈值
    thr_d=[95.762,92.330];     %斜线方向阈值
    thr_v=[99.321,94.122];     %垂直方向阈值
thr=[thr_h;thr_d;thr_v];
xd=wdencmp('lvd',x,'sym4',2,thr,'h');
%画出消噪后图像
subplot(224); image(xd); colormap(map);
title('独立阈值消噪图像');
axis square
```

输出结果(如图 2.70 所示):

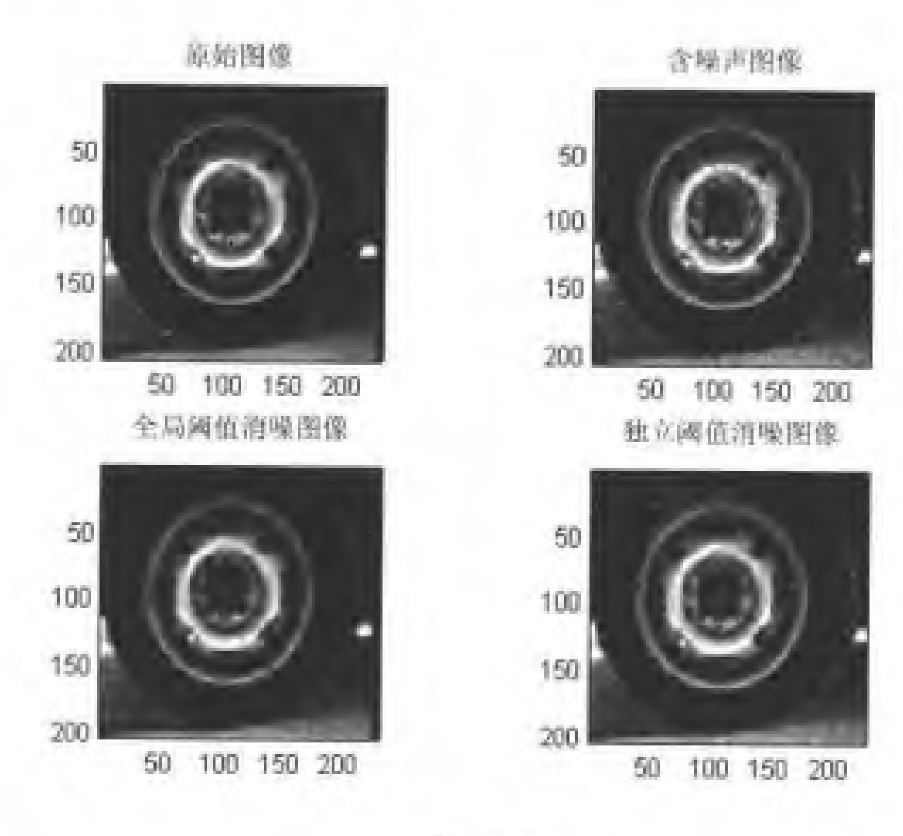


图　2.70

```
thr =
    93.0198
sorh =
s
keepapp =
     1
```

例 3：%装入原始图像，X 包含装入的图像

```
load woman;
x=X(100:200,100:200); nbc=size(map,1);
%画出原始图像
subplot(221); image(x); colormap(map);
title('原始图像');
axis square
%用小波 sym2 进行图像的压缩
[c,l]=wavedec2(x,5,'sym2');
thr=20;
[xd,cxd,lxd,perf0,perfl2]=wdencmp('gbl',c,l,'sym2',5,thr,'h',1);
%画出用全局阈值压缩图像
subplot(222); image(xd); colormap(map);
title('全局阈值压缩图像');
axis square
%此外，第一个选项可以使在不同的层用不同的阈值进行量化
%低频系数被保存，每层的阈值在三个方向上的值如下
    thr_h=[17,18];    %水平方向阈值
    thr_d=[19,20];    %斜线方向阈值
    thr_v=[21,22];    %垂直方向阈值
thr=[thr_h;thr_d;thr_v];
[xd,cxd,lxd,perf0,perfl2]=wdencmp('lvd',x,'sym8',2,thr,'h');
%画出用独立阈值压缩图像
subplot(223); image(xd); colormap(map);
title('独立阈值压缩图像');
axis square
```

结果输出(如图 2.71 所示)。

参见：ddencmp,wavedec,wavedec2,wden,wpdencmp,wthresh

5. wnoise

功能：产生含噪声的测试函数数据

格式：① X=wnoise(NUM,N)

② [X,XN]=wnoise(NUM,N,SNRAT)

③ [X,XN]=wnoise(NUM,N,SNRAT,INIT)

说明：该函数用来产生一个含噪声的测试函数数据，其中，NUM 为产生函数的类型，具体见表 2-5。N 表示采样点个数为 2^N 个，SNRAT 为信噪比，X 为含噪声的信号，XN 为不含噪声的信号。

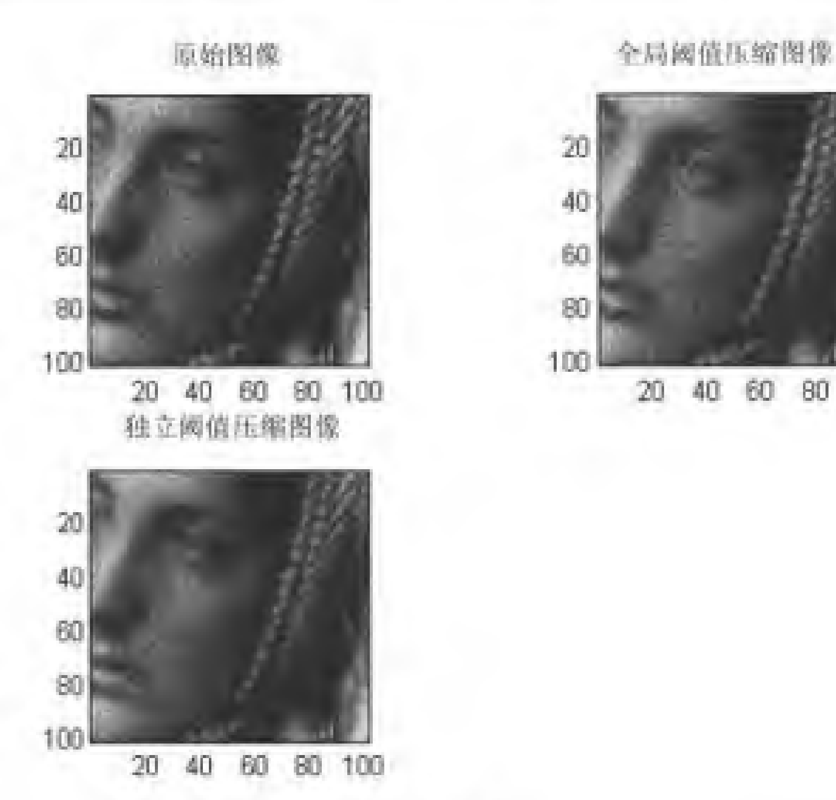


图　2.71

表 2-5　产生函数的类型

NUM	表示产生函数的类型
1	Blocks
2	Bumps
3	Heavy sine
4	Doppler
5	Quadchirp
6	Mishmash

举例：%产生一个采样点为 2^{10} 个的 Heavy sine 信号

```
x1=wnoise(3,10);
figure(1);
subplot(511); plot(x1); Ylabel('x1');
%产生个信噪比为 7，采样点的个数为 2^10的 Doppler 信号
[x2,noisyx2]=wnoise(4,10,7);
subplot(512); plot(x2); Ylabel('x2');
subplot(513); plot(noisyx2); Ylabel('noisyx2');
%产生一个 Doppler 信号
init=2055415866;
```

```
[x3,noisyx3]=wnoise(4,10,7,init);
subplot(514); plot(x3); Ylabel('x3');
subplot(515); plot(noisyx3); Ylabel('noisyx3');
%画出所有的测试函数
figure(2);
ind=linspace(0,1,2^10);
for i=1:6
    x=wnoise(i,10);
    subplot(6,1,i); plot(ind,x);
end
```

输出结果(如图 2.72 所示)。

参见：wden

6. wnoisest

功能：估计一维小波系数的标准偏差

格式：STDC=wnoisest(C,L,S)

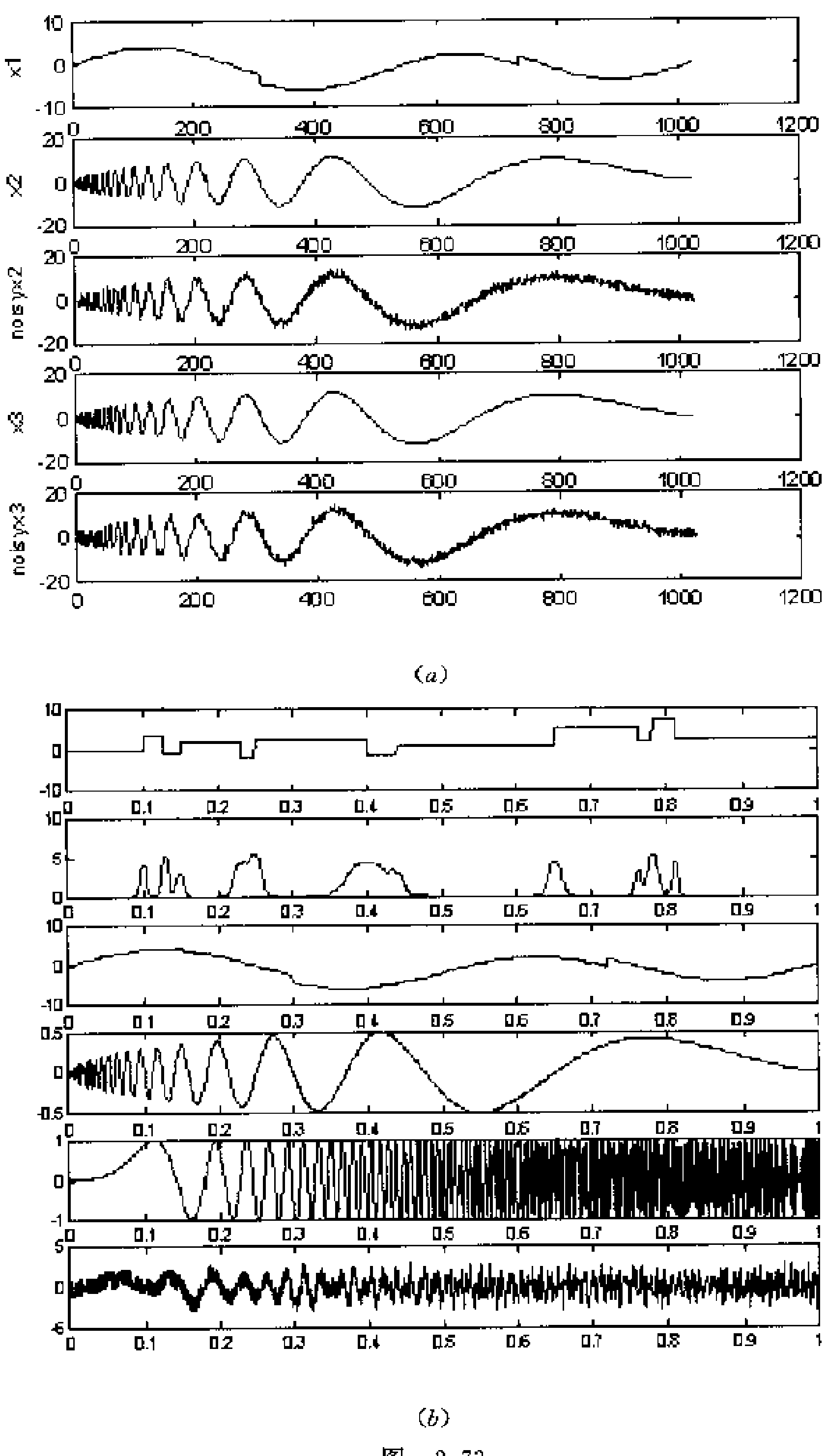
说明：对于输入信号向量 S 的各层节点，返回其高频系数的标准偏差的估计值，[C,L]是输入信号 S 的小波分解结构(参见 wavedec)。这种估计值采用 Maximum 的绝对偏差除以 0.6745，它在一维零均值高斯白噪声的信号模型中进行消噪处理时很有用。

举例：%产生高斯白噪声

```
init=2055415866; randn('seed',init);
x=randn(1,1000);
%用 db3 小波对信号进行 3 层分解
[c,l]=wavedec(x,2,'db3');
%估计第一层分解系数和第二层分解系数的标准偏差
%由于信号 x 是单位方差白噪声，所以估计很接近 1
pc1=wnoisest(c,l,1:2)
%假设 x 包含 10 个逸出值
ind=50:50:500; x(ind)=100*ones(size(ind));
%用 db3 小波对信号进行 1 层分解
[ca,cd]=dwt(x,'db3');
%对 cd 标准偏差的普通估计，它过高估计了噪声水平
pc2=std(cd)
%对 cd 标准偏差的鲁棒性估计，它使噪声水平接近 1
pc3=median(abs(cd))/0.6745
```

输出结果：

```
pc1 =
    1.0111    1.0763
pc2 =
```



(*a*)

(*b*)

图　2.72

    8.0206
pc3 =
    1.0540

参见：thselect，wavelet，wden

7. wpdencmp

功能：用小波包进行信号的消噪或压缩

格式：① [XD,TREED,DATAD,PERF0,PERFL2]＝wpdencmp(X,SORH,N,'wname',CRIT,PAR,KEEPAPP)

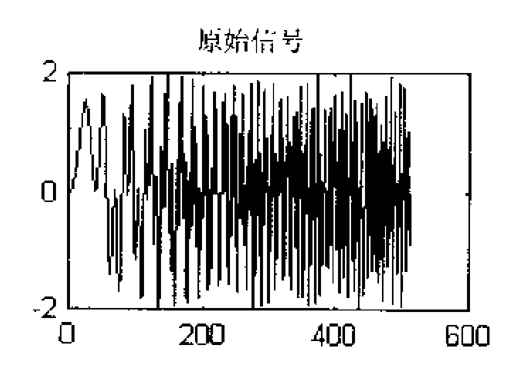
② [XD,TREED,DATAD,PERF0,PERFL2]＝wpdencmp(TREE,DATA, SORH,CRIT,PAR,KEEPAPP)

说明：wpdencmp 是一个一维或二维消噪或压缩函数。它用小波包对信号或图像进行消噪或压缩。用小波包进行消噪和压缩的思路和过程与用小波的一样(参见 wden 和 wdencmp)。格式①对输入信号 X(一维或二维)进行消噪或压缩后返回 XD(消噪或压缩后的结果)，消噪与压缩的过程也即系数经过阈值处理的过程。其中，'wname'指定所用的小波包函数，输出变量[TREED,DATAD]是小波包最佳分解结构(参见 besttree)。PERFL2 和 PERF0 是恢复和压缩 L^2 范数百分比，PERFL2＝100 * (XD 的小波包系数范数/X 的小波包系数范数)²。如果 X 是一个一维信号，小波'wname'是一个正交小波，则 $PERFL2=\frac{100\|XD\|^2}{\|X\|^2}$。N 表示小波分解的层数，'wname'是一个包含小波名的字符串，SORH('s'或'h')是软阈值或硬阈值的选择(参见 wthresh)。用熵标准来实现最佳分解，熵标准由字符串 CRIT 和参数 PAR 定义(参见 wentropy)。阈值参数也是 PAR。如果 KEEPAPP＝1，则低频系数不进行阈值量化(即系数不会受到改变)，反之，低频系数要进行阈值量化(即系数会受到改变)。

格式②和格式①有相同的输出参数和输入选项，只不过它是直接从小波包分解结构[TREE,DATA]中进行信号消噪或压缩处理(参见 maketree 和 wpdec)。

举例：

例 1：
```
%装入原始信号，x 包含装入的信号
load sumlichr; x=sumlichr;
%画出原始信号
subplot(221); plot(x);
title('原始信号');
%用 wdencmp 进行信号的压缩，并采用默认阈值(参见 ddencmp)
[thr,sorh,keepapp,crit]=ddencmp('cmp','wp',x)
%用全局阈值选项进行信号的压缩
[xc,treed,datad,perf0,perfl2]= …
wpdencmp(x,sorh,3,'db2',crit,thr,keepapp);
%画出压缩信号
subplot(222); plot(xc);
title('压缩后信号');
```
输出结果(如图 2.73 所示)。
```
thr =
    0.5193
sorh =
```



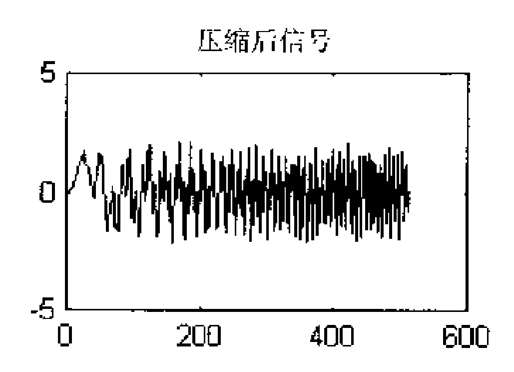


图　2.73

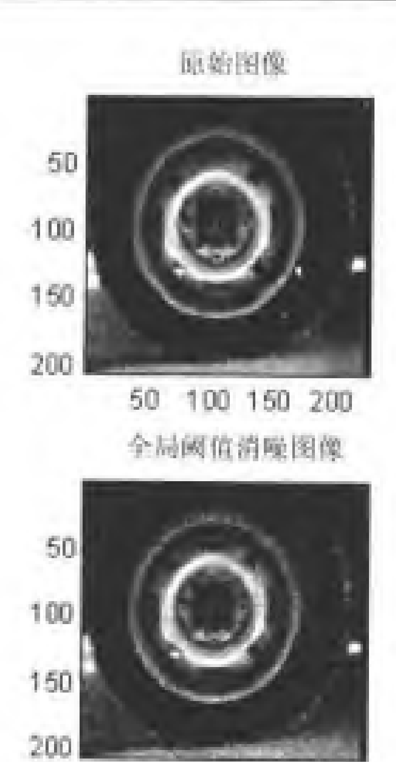
```
h
keepapp =
      1
crit =
threshold
```

例 2：%装入原始图像，X 包含装入的图像

```
load tire;
%画出原始图像
subplot(221); image(X); colormap(map);
title('原始图像');
axis square
%产生含噪图像
init=2055615866; randn('seed',init)
x=X+18 * randn(size(X));
%画出含噪声图像
subplot(222); image(x); colormap(map);
title('含噪声图像');
axis square
%用 wpdencmp 进行图像的消噪，采用默认阈值(参见 ddencmp)
[thr,sorh,keepapp,crit]=ddencmp('den','wp',x)
%用全局阈值选项进行图像的消噪
xd=wpdencmp(x,sorh,3,'sym4',crit,thr,keepapp);
%画出消噪后图像
subplot(223); image(xd); colormap(map);
title('全局阈值消噪图像');
axis square
```

输出结果(如图 2.74 所示)。

```
thr =
    5.1987
```



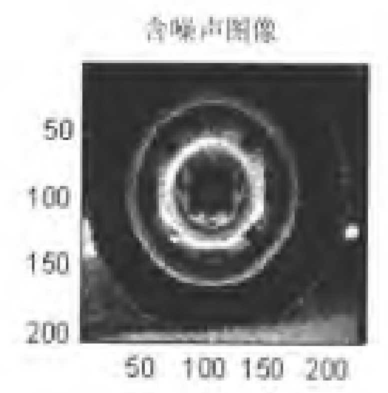


图 2.74

```
sorh =
h
keepapp =
      1
crit =
sure
```

例 3：%产生一个重正弦信号(heavy sine)，以及它的含噪信号

```
[xref,x]=wnoise(5,11,7,init);
xref=xref(1:1000);
x=x(1:1000);
%画出原始信号
subplot(221); plot(xref);
title('原始信号');
%画出含噪声信号
subplot(222); plot(x);
title('含噪声信号');
%用 wpdencmp 进行信号的消噪
n=length(x);
thr=sqrt(2*log(n*log(n)/log(2)));
xwpd=wpdencmp(x,'s',4,'sym4','sure',thr,1);
```

```
%画出 wpdencmp 消噪信号
subplot(223); plot(xwpd);
title('小波包消噪信号');
%下面用小波方法进行消噪，并把结果进行比较
xwd=wden(x,'rigrsure','s','one',4,'sym4');
%画出小波消噪信号
subplot(224); plot(xwd);
title('小波消噪信号');
```

输出结果(如图 2.75 所示)：

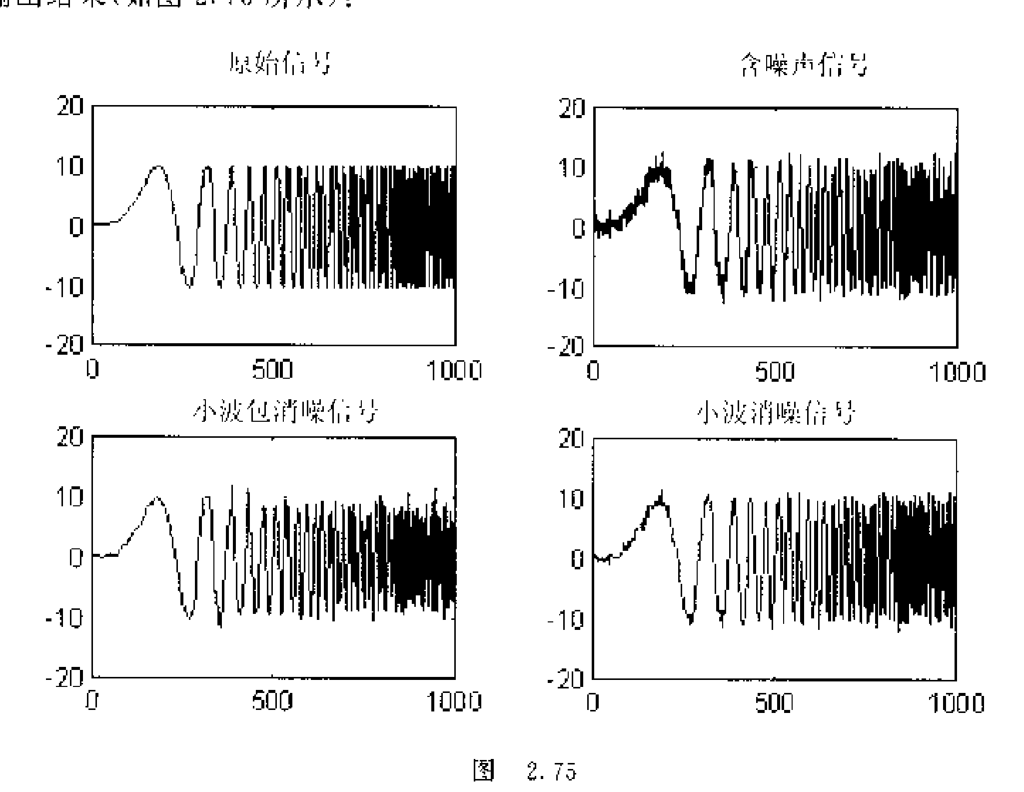


图 2.75

参见：ddencmp，wdencmp，wentropy，wpdec，wpdec2

8. wpthcoef

功能：进行小波包分解系数的阈值处理

格式：NDATA=wpthcoef(DATA，TREE，KEEPAPP，SORH，THR)

说明：wpthcoef 是一个一维或二维信号消噪和压缩的应用函数。它通过对信号的小波包分解结构[DATA，TREE](参见 maketree)进行系数的阈值处理后，返回一个新的数据结构 NDATA(小波包分解数据结构包含着小波包分解系数信息，当小波包分解系数发生变化时，小波包分解数据结构也发生相应的变化)。如果 KEEPAPP=1，则低频系数不进行阈值处理，否则，进行处理。如果 SORH='s'，则使用软阈值，如果 SORH='h'，则使用硬阈值。THR 是阈值大小，且是一个全局阈值。

举例：%装入一个原始图像

```
load woman;
```

```
x=X(100:200,100:200); nbc=size(map,1);
%画出原始图像
subplot(221); image(x); colormap(map);
title('原始图像');
axis square
%用 sym2 小波对图像进行 1 层分解
[t1,d1]=wpdec2(x,1,'sym2');
%设置一个全局阈值为 thr=-0.05
thr=-0.05;
%对图像分解系数(包括低频分解系数)进行软阈值量化，实现图像消噪
d1=wpthcoef(d1,t1,0,'s',thr);
%分别量化后的小波包分解系数进行重构
x1=wprcoef(t1,d1,1);
x2=wprcoef(t1,d1,2);
x3=wprcoef (t1,d1,3);
x4=wprcoef(t1,d1,4);
%合并重构后的图像
x5= x1+x2+x3+x4;
%画出合并重构后的图像
subplot(222); image(x5); colormap(map);
title('消噪后的图像');
axis square
```

输出结果(如图 2.76 所示)：

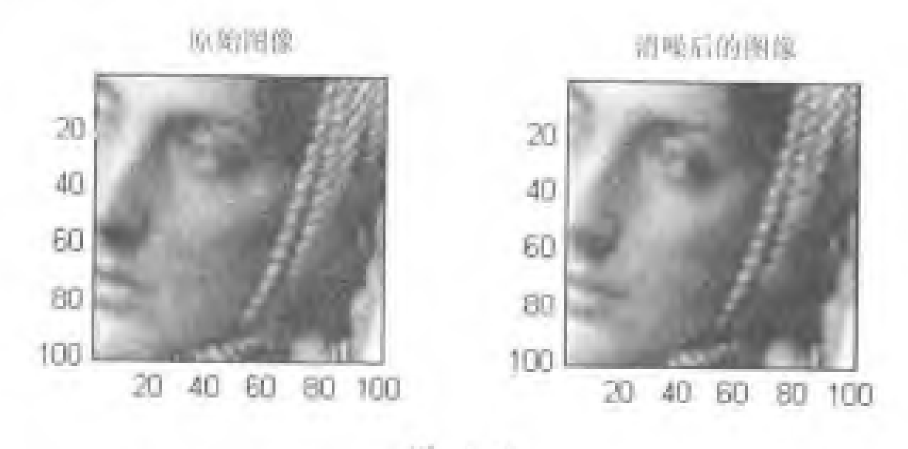


图 2.76

参见：maketree，wpdec，wpdec2，wpdencmp，wthresh

9. wthcoef

功能：一维信号的小波系数阈值处理

格式：① NC=wthcoef('d',C,L,N,P)

② NC=wthcoef('d',C,L,N)

③ NC=wthcoef('a',C,L)

④ NC=wthcoef('t',C,L,N,T,SORH)

说明：wthcoef是一个一维信号的消噪和压缩函数，它返回经过对小波分解结构[C,L]进行处理后返回新的小波分解向量NC，[NC，L]即构成一个新的小波分解结构。N是一个包含高频尺度的向量，P是一个包含把较小系数置0百分比信息的向量(该向量元素值的范围为0～100)，N和P必须是相同的长度，且向量N必须满足1<=N(i)<=length(L)－2。

格式①返回对小波分解结构[C,L](参见wavedec)进行比例压缩后的小波分解向量NC。其中，向量N定义待压缩的高频尺度，向量P定义把相应绝对值较小的系数置为0的百分比，

格式②返回小波分解结构[C，L]中，N所指定尺度的高频系数全部置0后的小波分解向量NC。

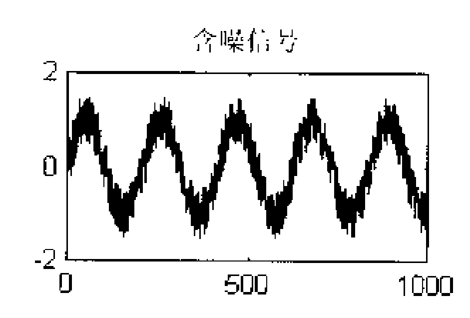
格式③返回把小波分解结构[C，L]中低频系数全部置0后的小波分解向量NC。

格式④返回对小波分解结构[C，L]经过阈值处理后的小波分解向量NC。N是一个包含高频尺度的尺度向量，T是与尺度向量N相对应的阈值向量，它定义每个尺度相应的阈值，N和T长度相等。参数SORH用来对阈值方式进行选择，当SORH='s'时，为软阈值，当SORH='h'时，为硬阈值。

举例：
```
%装入一个原始含噪信号
load noissin; s=noissin(1:1000);
%画出含噪信号的波形
subplot(321); plot(s);
title('含噪信号');
%用db3小波对信号进行3层小波分解
[c,l]=wavedec(s,3,'db3');
%设置尺度向量n
n=[1,2,3];
%设置阈值向量p
p=[98,99,97];
%对高频系数进行阈值处理
nc=wthcoef('d',c,l,n,p);
%对新的小波分解结构[nc,l]进行重构
ss=waverec(nc,l,'db3');
%画出重构后信号的波形
subplot(322); plot(ss);
title('消噪后的信号');
```

输出结果(如图2.77所示)。

参见：wavedec，wthresh



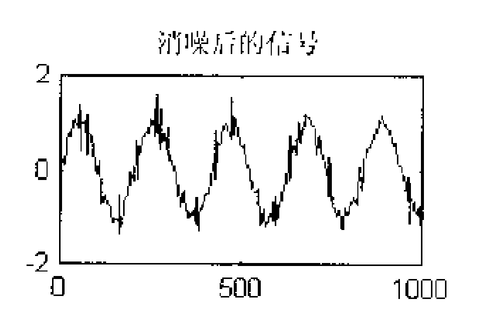


图 2.77

10. wthcoef2

功能：二维信号的小波系数阈值处理

格式：① NC=wthcoef2('type',C,S,N,T,SORH)

② NC=wthcoef2('type',C,S,N)

③ NC=wthcoef2('a',C,S)

④ NC=wthcoef2('t',C,S,N,T,SORH)

说明：wthcoef2 是一个二维信号的消噪和压缩函数。它返回经过对小波分解结构[C,S]进行处理后返回新的小波分解向量 NC，[NC，S]即构成一个新的小波分解结构。N 是一个包含高频尺度的向量，T 是相应的阈值，且向量 N 和 T 必须是相同的长度，向量 N 必须满足 1<=N(i)<=size(S,1)-2。

格式①通过对小波分解结构[C,S](参见 wavedec2)进行阈值处理后，返回'type'(水平、对角线或垂直)方向上的小波分解向量 NC。

对于格式②，'type'='h'(或'v'或'd')时，函数返回在 N 中定义的尺度的高频系数全部置零后的'type'方向系数。

格式③返回将低频系数全部置零后的系数。

格式④返回对小波分解结构[C，S]经过阈值处理后的小波分解向量 NC。N 是一个包含高频尺度的尺度向量，T 是与尺度向量 N 相对应的阈值向量，它定义每个尺度相应的阈值，N 和 T 长度相等。参数 SORH 用来对阈值方式进行选择，当 SORH='s'时，为软阈值，当 SORH='h'时，为硬阈值。

举例：%下面装载原始图像，X 中含有被装载的信号

```
load woman;
%画出原始图像
subplot(221); image(X); colormap(map);
title('原始图像');
axis square
%产生含噪图像
init=2055615866; randn('seed',init)
x=X+18 * randn(size(X));
%画出含噪声图像
```

```
subplot(222)；image(x)；colormap(map)；
title('含噪声图像')；
axis square；
%用小波函数 sym4 对 x 进行 2 层小波分解
[c,s]=wavedec2(x,2,'sym4')；
%设置尺度向量 n
n=[1,2]；
%设置阈值向量 p
p=[233,233]；
%对三个方向高频系数进行阈值处理
nc=wthcoef2('h',c,s,n,p,'s')；
nc=wthcoef2('v',c,s,n,p,'s')；
nc=wthcoef2('d',c,s,n,p,'s')；
%对新的小波分解结构[nc,s]进行重构
xx=waverec2(nc,s,'sym4')；
%画出重构后信号的波形
subplot(223)；image(xx)；
title('消噪后的图像')；
axis square；
```

输出结果(如图 2.78 所示)：



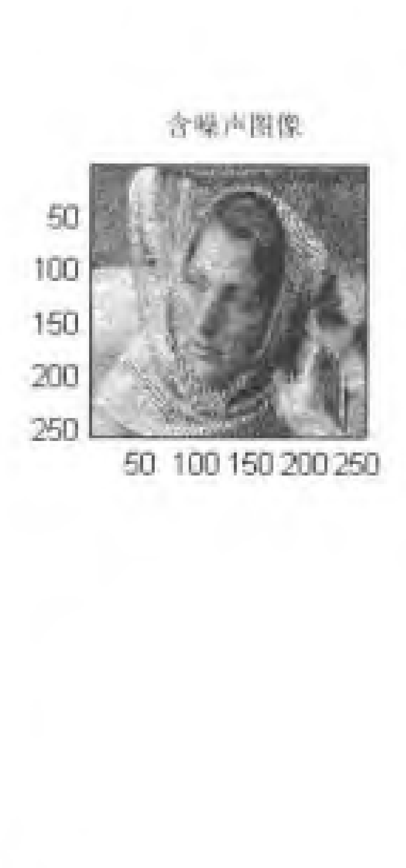


图　2.78

参见：wavedec2，wthresh

11. wthresh

功能：进行软阈值或硬阈值处理

格式：Y＝wthresh(X,SORH,T)

说明：该函数用于对信号 X 进行软阈值或硬阈值处理。向量 X 为待处理的信号，T 是阈值大小(T＞0)，SORH 参数用于软硬阈值的选择。

如果 SORH＝'s'，则为软阈值处理。对于软阈值处理，Y＝wthresh(X,'s',T)返回的是 $Y=\mathrm{SIGN}(X)\cdot(|X|-T)_+$，即把信号的绝对值与阈值进行比较，小于或等于阈值的点变为 0，大于阈值的点变为该点值与阈值的差值。

如果 SORH＝'h'则为硬阈值处理。对于硬阈值处理，Y＝wthresh(X,'h',T)返回的是 $Y=X\cdot 1_{(|X|>T)}$，即把信号的绝对值与阈值进行比较，小于或等于阈值的点变为 0，大于阈值的点保持不变。一般说来，用硬阈值处理后的信号比用软阈值处理后的信号更为粗糙。

举例：%程序清单

```
y=linspace(-1,1,100); %产生一个在[-1，1]之间的斜坡信号
subplot(231); plot(y);
title('原始信号');grid;
thr=0.4; %设定阈值
ythard=wthresh(y,'h',thr); %用硬阈值进行处理
subplot(232); plot(ythard);
title('硬阈值信号');grid;
ytsoft=wthresh(y,'s',thr); %用软阈值进行处理
suhplot(233); plot(ytsoft);
title('软阈值信号');grid;
```

输出结果(如图 2.79 所示)：

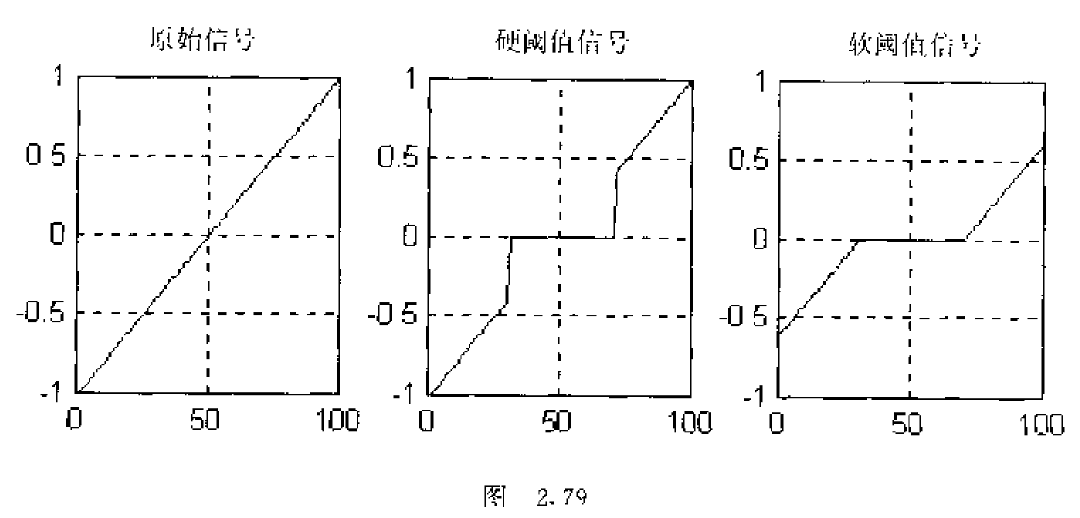


图　2.79

从上面的图形比较，我们可以明显看出而软阈值是对硬阈值的扩展，它首先令绝对值小于阈值的元素等于 0，然后将其余非零的元素向零进行缩小。硬阈值在 y＝±T 处不连

续，而软阈值在 y=±T 处是连续的。

参见：wden，wdencmp，wpdencmp

2.7 树操作应用函数

在小波包分解中，树结构(简称树)是一个非常重要的概念，树中的每一个结点即对应一个小波包，从某种意义上讲，对树的操作也就是对小波包的操作。在本节里，我们将详细讲述有关树操作应用函数的用法。

1. maketree

功能：创建树

格式：① [T,NB]=maketree(ORD,D)

② [T,NB]=maketree(ORD,D,NBI)

③ [T,NB]=maketree(ORD)

说明：maketree 是一个树操作应用函数。它创建一个叉数为 ORD，深度为 D 的树。ORD 是一个整数，它等于一个非终结点所有的子结点个数，即每一个非终结点都有 ORD 子结点。对于小波包分解，最简单的是在一维情况下的二叉树结构(即 ORD=2 的情况)，以及在二维情况下的四叉树结构(即 ORD=4 的情况)。

格式①返回一个叉数为 ORD，深度为 D 的树结构。输出变量 NB 是终结点的个数(NB=ORD^D)，输出向量 T 是如下的形式：T=[T(1)…T(NB+1)]，其中 T(i)，(i=1，…，NB)是终结点的索引，且具有 T(NB+1)=−ORD。

结点是从左到右，从上到下编号的，根结点的索引为 0。当用三个输入变量时，即对于格式②，函数返回 T，T 是一个(NBI+1)×(NB+1)矩阵，且 T(1，:)与格式①中的向量相同。在 T 为 T(2:NBI+1，:)的范围内(即在矩阵的第二行到第 NBI 行内)，用户可以加入自己的内容。格式①等价于[T,NB]=maketree(ORD,D,0)。格式③等价于[T,NB]=maketree(ORD,0,0)。

举例：%创建一个叉数为 2，深度为 3 的二叉树

```
t2=maketree(2,3);
plottree(t2) %画树结构 t2 的图形
%创建一个叉数为 4，深度为 2 的四叉树
t4=maketree(4,2);
plottree(t4) %画树结构 t4 的图形
```

输出结果(如图 2.80、2.81 所示)。

参见：plottree，wtreemgr

2. plottree

功能：画树结构图形

格式：plottree(T)

说明：plottree 是一个图形化树操作应用函数，它可以画出树结构 T 的图形(参见

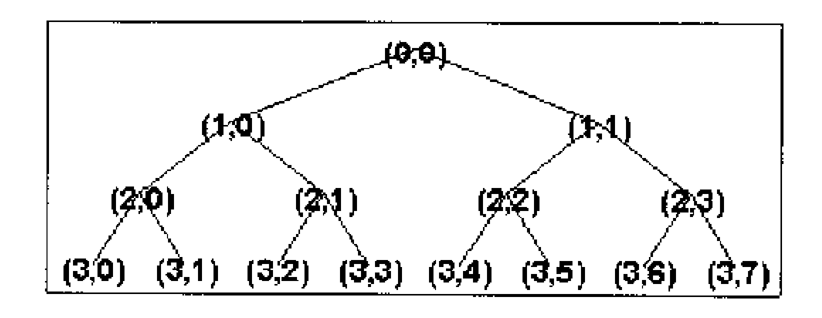


图 2.80　完整二叉树的结构图

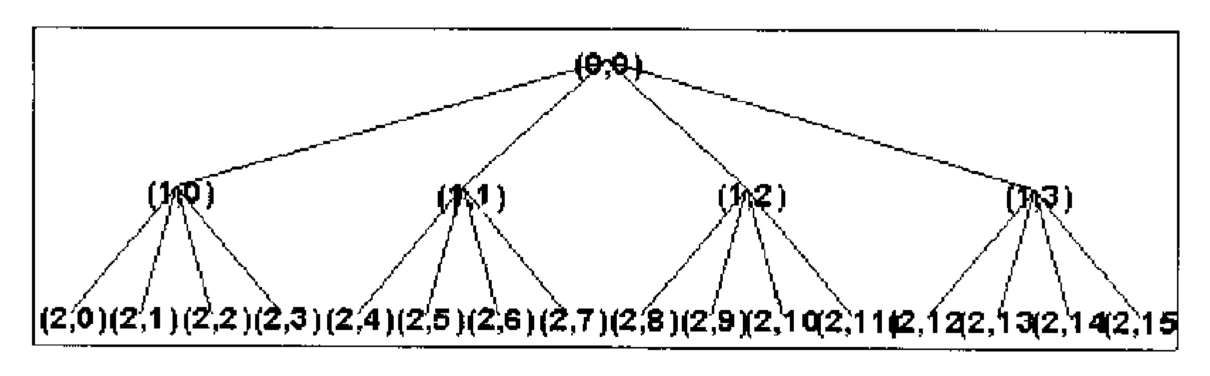


图 2.81　完整四叉树的结构图

maketree)。

举例：%创建一个叉数为 2，深度为 3 的二叉树

```
t=maketree(2,3);
plottree(t) %画树结构 t 的图形
```

输出结果(如图 2.82 所示)：

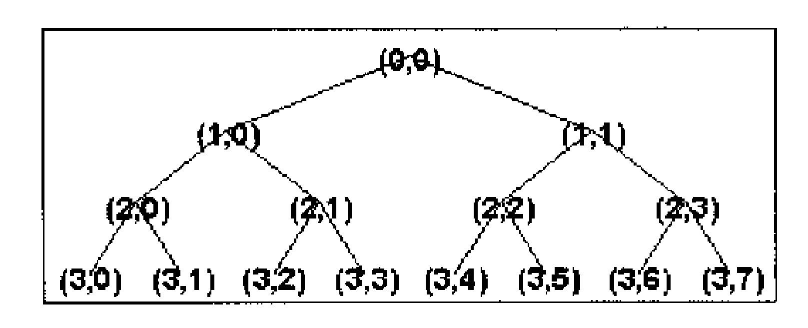


图　2.82

参见：maketree，wpdec，wpdec2

3. nodejoin

功能：重组结点

格式：① T=nodejoin(T,N)

② T=nodejoin(T)

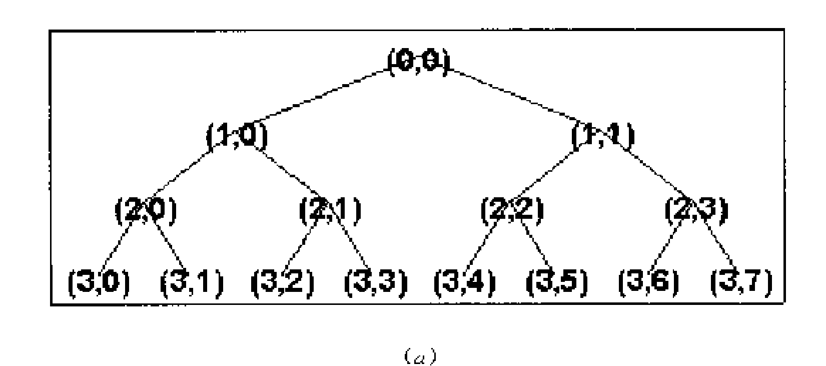
说明：nodejoin 是一个树操作应用函数。格式①对结点 N 进行重组，返回修改后的树结构

T；格式②等价于 T=nodejoin(T,0)。树结点是从左到右，从上到下编号的，根结点的索引为 0。

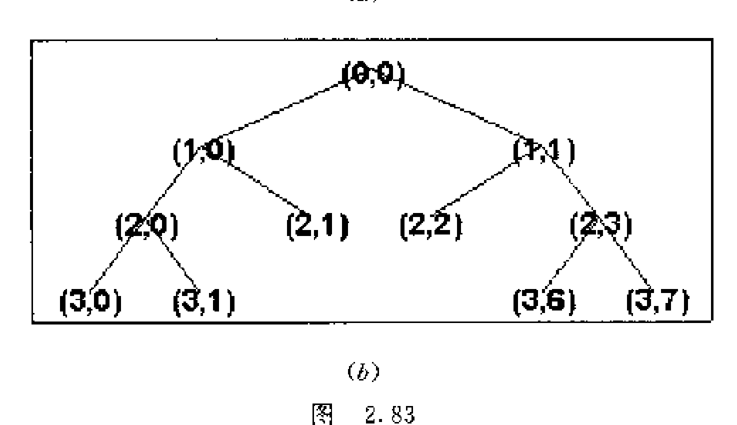
举例：%创建一个叉数为 2，深度为 3 的二叉树

```
t=maketree(2,3);
plottree(t) %画树结构 t 的图形
%合并结点 4 和 5
tt=nodejoin(t,5);
tt=nodejoin(tt,4);
%画出合并后的树结构图
plottree(tt)
```

输出结果(如图 2.83 所示)：



(a)



(b)

图 2.83

参见：maketree,nodesplt,wtreemgr

4. allnodes

功能：计算树结点

格式：① N=allnodes(T)

② N=allnodes(T,'deppos')

说明：allnodes 是一个树操作应用函数，它可以返回两种树结点描述形式中的任何一种，即要么是树的索引形式，要么是树的深度—位置形式。树结点是从左到右，从上到下编号的，根结点的索引为 0。格式①返回列向量 N，它包括树结构 T 的所有结点的索引。格式②返回的是 $M\times2$(M 为结点的总数)矩阵 N，N 指明所有结点的深度和位置，其中，第一列

N(i,1)是结点 i 的深度，第二列 N(i,2)是结点 i 的位置。

举例：%创建初始树，树的叉数(也称阶数)为 2

```
ord=2;
t=maketree(ord,3); %二叉树的深度为 3
tt=nodejoin(t,5);
tt=nodejoin(tt,4);
plottree(tt)
%以索引形式列出 tt 结点
aln_ind=allnodes(tt)
%以深度—位置形式列出 tt 结点
aln_depo=allnodes(tt,'deppos')
```

输出结果(如图 2.84 所示)：

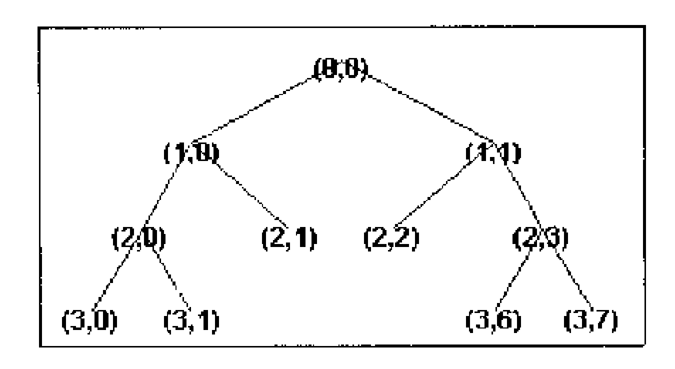


图　2.84

```
aln_ind =
     0
     1
     2
     3
     4
     5
     6
     7
     8
    13
    14

aln_depo =
     0     0
     1     0
     1     1
```

```
2    0
2    1
2    2
2    3
3    0
3    1
3    6
3    7
```

参见：maketree

5. depo2ind

功能：将深度－位置结点形式转化成索引结点形式

格式：N＝depo2ind(ORD,[D,P])

说明：depo2ind 是一个树操作应用函数。对一个叉数为 ORD 的树，它计算出与[D，P]相对应的结点的索引向量 N，结点的深度和位置在[D,P]中编码。D,P 和 N 都是列向量，其值的限制为

- D 表示深度，且 $0 \leqslant D \leqslant d\ max$
- P 表示深度 D 处的位置，且 $0 \leqslant P \leqslant order^{D-1}$
- N 表示索引，且 $0 \leqslant N < (order^{(d\ max+1)}-1)/(order-1)$

注意，对一个列向量 X，我们有 depo2ind(0,X)＝X

举例：

```
ord=2; %创建初始树
t=maketree(ord,3); %二叉树的深度为3
%进行树结点的合并
tt=nodejoin(t,5);
tt=nodejoin(tt,4);
plottree(tt)
%以深度和位置形式列出tt结点
aln_depo=allnodes(tt,'deppos')
%将深度－位置结点形式转化成索引形式
aln_ind=depo2ind(ord,aln_depo)
```

输出结果(如图 2.85 所示)：

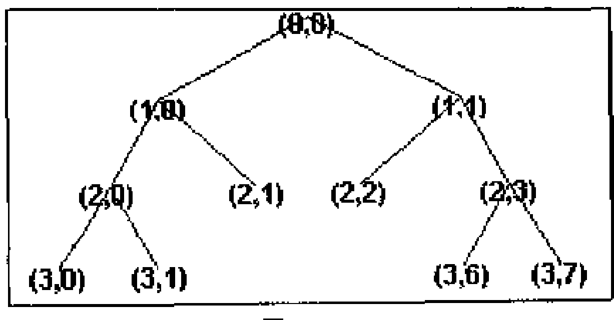


图　2.85

```
aln_depo =
     0     0
     1     0
     1     1
     2     0
     2     1
     2     2
     2     3
     3     0
     3     1
     3     6
     3     7

aln_ind =
     0
     1
     2
     3
     4
     5
     6
     7
     8
    13
    14
```

参见：ind2depo，maketree，wtreemgr

6. ind2depo

功能：将索引结点形式转化成深度－位置结点形式

格式：N＝ind2depo(ORD，N)

说明：ind2depo 是一个树操作应用函数。对一个叉数为 ORD 的树，它计算出与索引向量 N 相对应的结点的深度 D 和该深度上的位置 P，即返回一个 $M\times 2$ 的矩阵（M 为结点总数）。结点是从左到右，从上到下编号的，根结点的索引为 0。D，P 和 N 都是列向量，其值的限制为：

- D 表示深度，且 $0\leqslant D\leqslant d\ max$
- P 表示深度 D 处的位置，且 $0\leqslant P\leqslant order^{D-1}$
- N 表示索引，且 $0\leqslant N<(order^{(d\ max-1)}-1)/(order-1)$

注意，[D，P]＝ind2depo(ORD，[D，P])

举例：%创建初始树

```
ord=2;
t=maketree(ord,3); %二叉树的深度为 3
tt=nodejoin(t,5);
tt=nodejoin(tt,4);
plottree(tt)
%以索引形式列出 tt 结点
aln_ind=allnodes(tt)
%将索引形式转化成深度-位置结点形式
[depth,pos]=ind2depo(ord,aln_ind);
aln_depo=[depth,pos]
```

输出结果(如图 2.86 所示)：

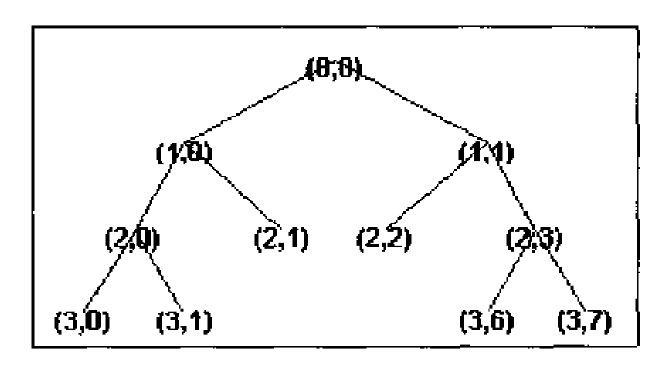


图 2.86

```
aln_ind =
     0
     1
     2
     3
     4
     5
     6
     7
     8
     13
     14

aln_depo =
     0     0
     1     0
     1     1
```

```
2    0
2    1
2    2
2    3
3    0
3    1
3    6
3    7
```

参见：depo2ind，maketree，wtreemgr

7. isnode

功能：判断结点是否存在

格式：R=isnode(T,N)

说明：isnode 是一个树操作应用函数，如果结点 N 在树结构 T 中，则函数返回 1(表示该结点在结构树中存在)，否则返回 0(表示该结点在结构树中不存在)。N 可以是一个列向量，包含结点的索引，也可以是一个 $M\times 2$ 的矩阵(M 为结点总数)，包含结点的深度和位置信息。当 N 为矩阵时，N(i,1)是第 i 个结点的深度，N(i,2)是第 i 个结点的位置。结点是从左到右，从上到下编号的，根结点的索引为 0。

举例：%创建初始树

```
ord=2;
t=maketree(ord,3); %二叉树的深度为 3
tt=nodejoin(t,5);
tt=nodejoin(tt,4);
plottree(tt)
%检查结点的索引
index=isnode(tt,[1;3;25])
%检查结点的深度—位置
dep_pos=isnode(tt,[1,0;3,1;4,5])
```

输出结果(如图 2.87 所示)：

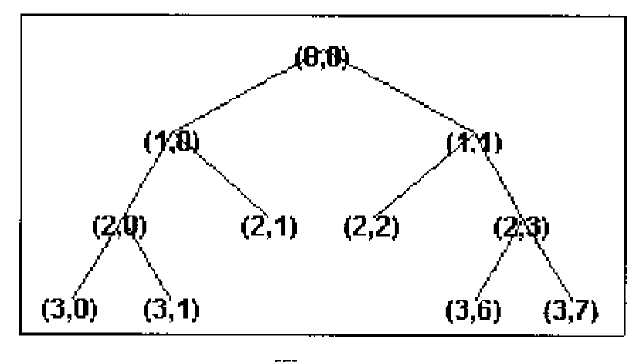


图　2.87

```
index =
```

```
        1
        1
        0

dep_pos =
        1
        1
        0
```

参见：istnode，maketree，wtreemgr

8. istnode

功能：判断结点是否是终结点并返回排列值

格式：R=istnode(T,N)

说明：istnode 是一个树操作应用函数。如果结点 N 是树结构 T 的终结点，则函数按照从左到右的终结点顺序返回终结点的排列，否则返回 0(表示该结点不是终结点)。N 可以是一个列向量，包含结点的索引，也可以是一个矩阵，包含结点的深度和位置信息。当 N 为矩阵时，N(i,1)是第 i 个结点的深度，N(i,2)是第 i 个结点的位置。结点是从左到右，从上到下编号的，根结点的索引为 0。

举例：%创建初始树

```
ord=2;
t=maketree(ord,3); %二叉树的深度为 3
tt=nodejoin(t,5);
tt=nodejoin(tt,4);
plottree(tt)
%寻找终结点，返回终结点在树结构中的索引
a1=istnode(tt,[14])
a2=istnode(tt,[15])
a3=istnode(tt,[1;7;14;25])
a4=istnode(tt,[1,0;3,1;4,5])
```

输出结果(如图 2.88 所示)：

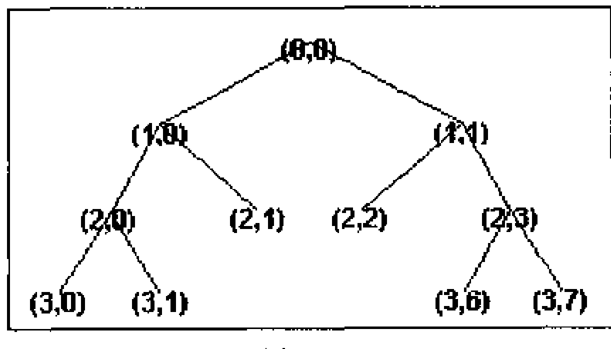


图　2.88

```
a1 =
     6
a2 =
     0
a3 =
     0
     1
     6
     0
a4 =
     0
     2
     0
```

参见：isnode，maketree，wtreemgr

9. nodeasc

功能：计算上溯结点

格式：① D=nodeasc(T,N)

② D=nodeasc(T,N,'deppos')

说明：nodeasc 是一个树操作应用函数。它返回树结构 T 中结点 N 的所有上溯结点的索引，其中，N 可以是结点的索引形式(即为一个向量)，也可以是结点的深度－位置形式(即为一个矩阵)。对于格式①，返回参数 T 是一个列向量；对于格式②，返回参数 T 是一个包含所有上溯结点的深度和位置信息的矩阵，其中，D(i,1)是第 i 个上溯结点的深度，D(i,2)是第 i 个上溯结点的位置。

举例：

```
%创建初始树
ord=2;
t=maketree(ord,3); %二叉树的深度为 3
tt=nodejoin(t,5);
tt=nodejoin(tt,4);
plottree(tt)
%计算上溯结点
a1=nodeasc(tt,2)
a2=nodeasc(tt,2,'deppos')
a3=nodeasc(tt,[2 2])
a4=nodeasc(tt,[2,2],'deppos')
```

输出结果(如图 2.89 所示)。

```
a1 =
     2
     0
```

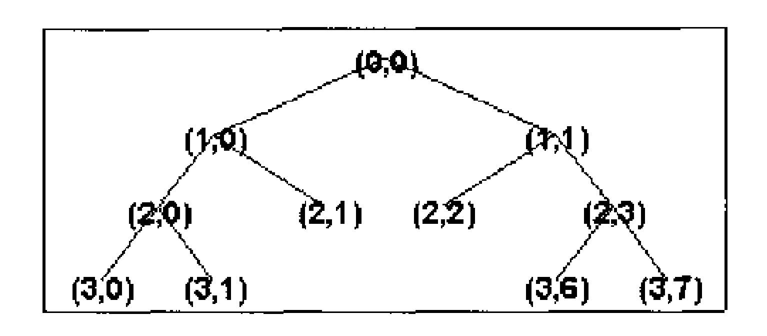


图　2.89

```
a2 =
     1     1
     0     0

a3 =
     5
     2
     0

a4 =
     2     2
     1     1
     0     0
```

参见：maketree，nodedesc，nodepar，wtreemgr

10. nodedesc

功能：计算下溯结点(子结点)

格式：① D=nodedesc(T,N)

② D=nodedesc(T,N,'deppos')

说明：nodedesc是一个树操作应用函数。它返回在树结构T中的结点N的所有子结点的索引，其中，N可以是结点的索引形式，也可以是结点的深度—位置形式。对于格式①，D是一个列向量，并且D(1)是结点N的索引；对于格式②，它返回的是一个包含所有下溯结点深度和位置信息的矩阵，其中，D(i,1)是第i个下溯结点的深度，D(i,2)是第i个下溯结点的位置。注意，返回的子结点包括结点本身。

举例：%创建初始树

```
ord=2;
t=maketree(ord,3); %二叉树的深度为3
tt=nodejoin(t,5);
tt=nodejoin(tt,4);
```

```
plottree(tt)
%计算下溯结点
a1=nodedesc(tt,2)
a2=nodedesc(tt,2,'deppos')
a3=nodedesc(tt,[2,2],'deppos')
a4=nodedesc(tt,[1,1],'deppos')
a5=nodedesc(tt,[1,1])
```

输出结果(如图 2.90 所示):

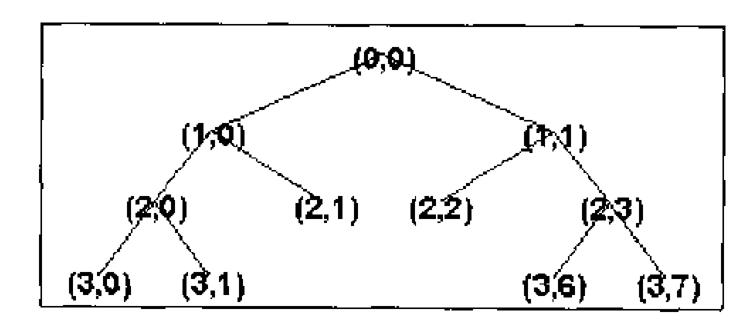


图 2.90

```
a1 =
     2
     5
     6
    13
    14

a2 =
     1     1
     2     2
     2     3
     3     6
     3     7

a3 =
     2     2

a4 =
     1     1
     2     2
     2     3
     3     6
```

```
        3       7

    a5 =
        2
        5
        6
       13
       14
```

参见：maketree，nodeasc，nodepar，wtreemgr

11. nodepar

功能：寻找父结点

格式：① F＝nodepar(T，N)

② F＝nodepar(T，N，'deppos')

说明：nodepar 是一个树操作应用函数。它返回在树结构 T 中，结点 N 的父结点索引。N 可以是一个包括索引信息的向量，也可以是一个包括深度—位置信息的矩阵。当 N 是矩阵时，N(i，1)是第 i 个结点的深度，N(i，2)是第 i 个结点的位置。对于格式①，它返回的是一个包括父结点索引信息的向量；对于格式②，它返回的是一个包括父结点深度－位置信息的矩阵，其中，F(i，1)是第 i 个结点的深度，F(i，2)是第 i 个结点的位置。

另外，nodepar(T，0)或 nodepar(T，[0，0])返回－1；nodepar(T，0，'deppos')或 nodepar(T，[0，0]，'deppos')返回[－1，0]。

举例：%创建一个叉数为 2，深度为 3 的二叉树

```
t=maketree(2,3);
%合并结点 4 和 5
tt=nodejoin(t,5);
tt=nodejoin(tt,4);
%画出合并后的树结构图
plottree(tt)
%寻找父结点
a1=nodepar(tt,[2,2],'deppos')
a2=nodepar(tt,[1;7;14])
```

输出结果(如图 2.91 所示)。

```
a1 =
      1       1

a2 =
      0
      3
      6
```

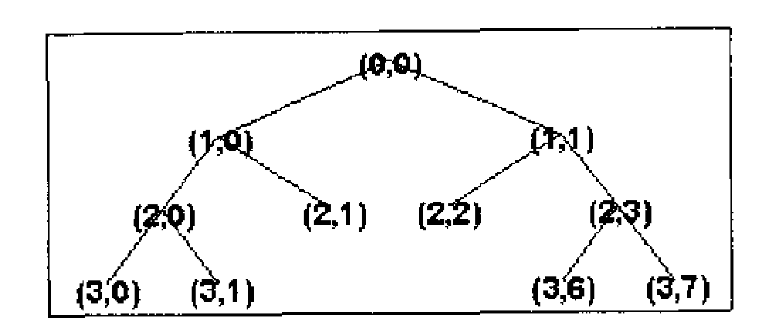


图 2.91

注意：父结点和上溯结点不是一个概念，父结点指的是与某结点直接相连的上一层结点，而上溯结点指的是与某结点相连的所有上层结点。

参见：maketree，nodeasc，nodedesc，wtreemgr

12. ntnode

功能：求终结点的个数

格式：NB=ntnode(T)

说明：ntnode 是一个树操作应用函数。它返回树结构 T 中所有终结点的个数。

举例：%创建一个叉数为 2，深度为 3 的二叉树

```
t=maketree(2,3);
plottree(t) %画树结构 t 的图形
ntnode(t) %求出终结点的个数
```

输出结果(如图 2.92 所示)：

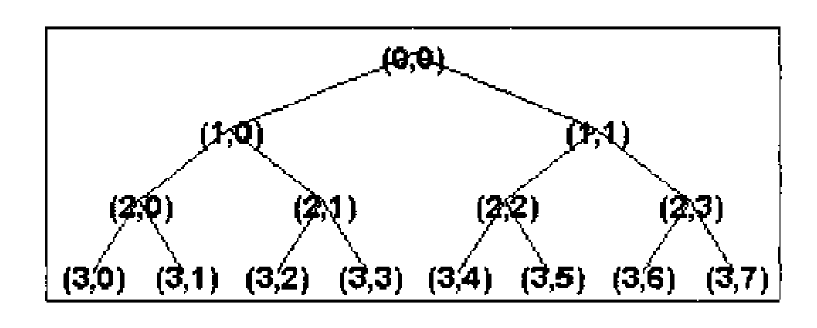


图 2.92

```
ans=
     8
```

参见：maketree，wtreemgr

13. nodesplt

功能：分割(分解)结点

格式：T=nodesplt(T,N)

说明：nodesplt 是一个树操作应用函数，它对结点 N 进行分解后，返回修改后的树结构 T。

举例：%创建一个叉数为 2，深度为 3 的二叉树

```
t=maketree(2,3);
plottree(t) %画树结构 t 的图形
%分解索引为 10 的结点
tt=nodesplt(t,10);
plottree(tt) %画新的树结构 tt 的图形
```

输出结果(如图 2.93、2.94 所示):

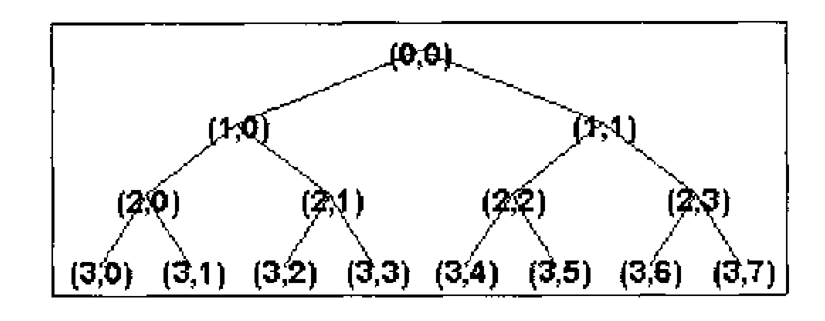


图 2.93　原始树图结构

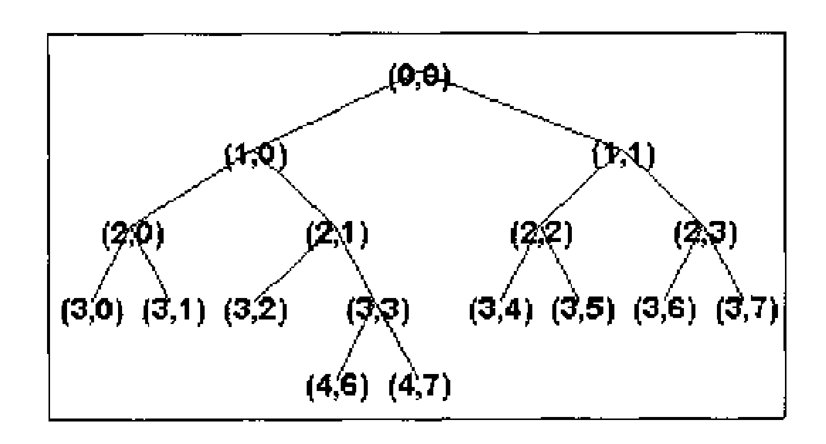


图 2.94　分解后的树结构

参见:maketree,nodejoin,wtreemgr

14. treedpth

功能:求树的深度

格式:D=treedpth(T)

说明:treedpth 是一个树操作应用函数。它返回树结构 T 的深度 D。

举例:%创建一个叉数为 2,深度为 3 的二叉树

```
t=maketree(2,3);
plottree(t)   %画树结构 t 的图形
treedpth(t)   %求出树结构的深度
```

输出结果(如图 2.95 所示):

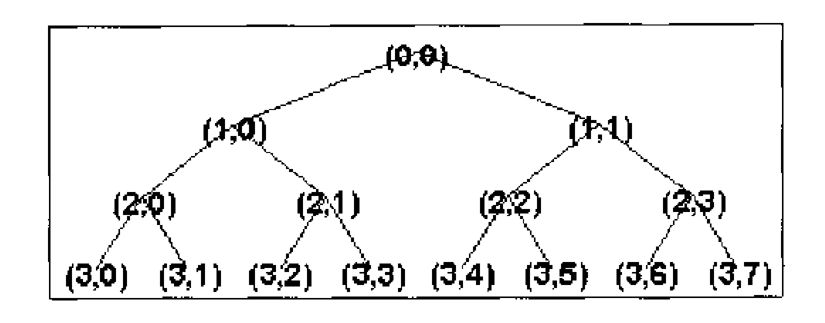


图 2.95

ans =

3

参见：maketree，wtreemgr

15. treeord

功能：求树结构的叉数

格式：ORD=treeord(T)

说明：treeord 是一个树操作应用函数，它返回树结构 T 的叉数 ORD。

举例：%创建一个叉数为 2，深度为 3 的二叉树

t=maketree(2,3)；

plottree(t) %画树结构 t 的图形

treeord(t) %求出树结构的叉数

输出结果(如图 2.96 所示)：

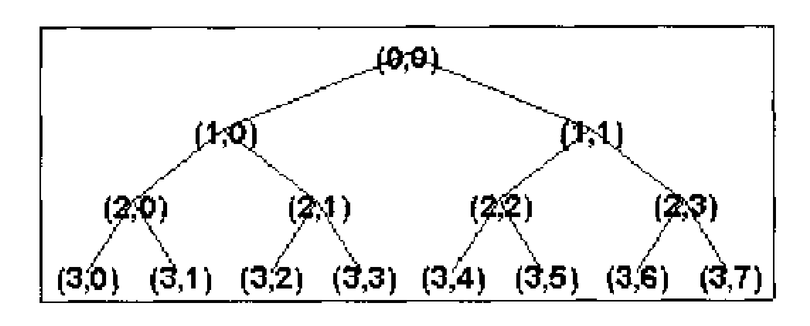


图 2.96

ans =

2

参见：maketree，wtreemgr

16. wdatamgr

功能：管理数据结构

格式：[OUT1,OUT2]=wdatamgr(OPT,D,IN3,IN4,IN5)

说明：wdatamgr 是一个树操作应用函数，D 是数据结构，OPT 是一个字符串选项，不同的选择具有不同的功能，具体如下：

· 'write_cfs'：向终结点写系数

data＝wdatamgr('write_cfs',data,tree,node,coefs)；

· 'read_cfs'：读取终结点的系数

coefs＝wdatamgr('read_cfs',data,tree,node)；

· 'read_ent'：读取结点的熵值向量。data是小波包分解的数据结构；nodes表示结点，当它以索引形式表示时，nodes是一个向量，当它以深度—位置形式表示时，nodes是一个$M\times 2$矩阵(M为结点的总个数)。

ent＝wdatamgr('read_ent',data,nodes)；

· 'read_ent0'：读取优化后的熵向量

ento＝wdatamgr('read_ento',data,nodes)；

· 'read_tp_ent'：读取熵的类型和参数

[type_ent,param]＝wdatamgr('read_tp_ent',data)；

· 'read_wave'：读取小波名

wave＝wdatamgr('read_wave',data)；

举例：%装入信号

```
load noisdopp; x=noisdopp;
%用shannon熵、db1小波对信号x进行三层小波包分解
[t,d]=wpdec(x,3,'db1','shannon');
%读取熵名
ent_name=wdatamgr('read_tp_ent',d)
%读取小波名
wave_name=wdatamgr('read_wave',d)
%读取小波包[3,2]的系数
cfs=wdatamgr('read_cfs',d,t,[3,2]);
%读取小波包[3,2]的熵和优化熵
ind_node=depo2ind(2,[3,2]);
ent=wdatamgr('read_ent',d,ind_node)
%优化熵是NaN，原因是熵没有被优化
ento=wdatamgr('read_ento',d,ind_node)
%修改小波包[3,2]的系数
ncfs=cos(cfs);
%更新小波包[3,2]的系数
d=wdatamgr('write_cfs',d,t,[3,2],ncfs);
%更新结点的熵值
d=entrupd(t,d,'shannon');
nent=wdatamgr('read_ent',d,ind_node)
```

输出结果：

```
ent_name =
shannon
```

```
wave .name =
db1

ent =
  -318.4298

ento =
     NaN

nent =
     22.2830
```

参见：wpdec，wpdec2，wtreemgr

17. wtreemgr

功能：管理树结构

格式：[OUT1,OUT2,OUT3,OUT4]=wtreemgr(OPT,STRUCTURE,IN3,IN4,IN5)

说明：wtreemgr 是一个树操作应用函数。OPT 允许的参数以及相关用途请见下面列出的各个函数：

- 'allnodes'：列出所有结点
- 'isnode'：检查结点是否存在
- 'istnode'：检查是否是终结点
- 'create'：创建一个树
- 'nodeasc'：计算上溯结点
- 'nodedesc'：寻找结点的所有(下溯结点)子结点
- 'nodepar'：寻找父结点
- 'ntnode'：求终结点的个数
- 'tnodes'：求终结点
- 'order'：求树的阶数
- 'depth'：求树的深度

树结构的创建请参见 maketree。

参 见：allnodes，isnode，istnode，maketree，nodeasc，nodedesc，nodepar，ntnode，tnodes，treedpth，treeord

2.8　如何添加自己的小波函数

一个信号的傅里叶变换其实就是该信号在一组正交的正弦函数($\sin \omega x$)和余弦函数($\cos \omega x$)上的投影，而一个信号的小波变换是它在一组小波函数簇上的投影。选用恰当的小波函数簇，可以很好地分析信号的特征，相反，若小波函数簇选取不正确，对信号进行

小波变换之后，信号在小波函数簇上的投影系数很可能淹没信号的特征。一组小波函数簇可以由一个小波基函数通过适当的尺度变换和迭代运算来产生，一般我们称这个小波基函数为小波函数。小波函数的构造以及选用也是当今小波理论研究的一个热点。在MATLAB中，系统只提供了八种小波函数，而从实际应用的角度来说，这是远远不够的，因为对于一个具体待分析的信号，往往某个小波函数对于信号的分析具有最佳的效果。在本节里，我们将详细讲述如何添加自己的小波函数。

2.8.1 如何准备添加一个小波函数

在MATLAB中，我们可以用wavemngr函数去添加一个小波函数，然而，在添加一个小波函数之前，必须进行如下的工作：

(1) 选择一个小波函数的全称名(full name)。小波函数的全名必须是一个字符串，如在MATLAB中已有的：Haar、Daubechies、Coiflets、Symlets、Morlet等。

(2) 选择一个小波函数的缩写名(short name)。缩写名必须是一个字符串，且其字符个数小于或等于4。如在MATLAB中已有的：haar、db、coif、sym、morl等。

(3) 选择一个小波类型。小波的类型通常具有四种情况。

① 具有有限冲激响应滤波器(IIR)的正交小波。这种小波函数可以通过尺度滤波器 w 进行定义，在MATLAB中，已经定义的此类函数有：Haar，Daubechies，Coiflets和Symlets。

② 具有有限冲激响应滤波器的双正交小波。这种小波函数可以通过尺度滤波器 wr(用于小波重构)和 wd(用于小波分解)进行定义，在MATLAB中，已经定义的此类函数有BiorSplines。

③ 不具有有限冲激响应滤波器但有尺度函数的正交小波。这种小波函数可以通过小波函数和尺度函数进行定义，在MATLAB中，已经定义的此类函数有：Meyer。

④ 不具有有限冲激响应滤波器也不具有尺度函数的正交小波。这种小波函数可以通过小波函数和尺度函数进行定义，在MATLAB中，已经定义的此类函数有：Morlet和Mexican hat。

(4) 定义一个在小波函数系列中的序列号。如果存在一个小波函数系，则小波函数系是由小波函数缩写名和一个序列号一起构成一个波函数名。字符串('nums')包含的是一个小波函数系列的所有序列号，各个序列号之间用空格隔开。对于③、④类型的小波函数和只有单个小波函数的情况，没有序列号参数。

例：对于Daubechies小波函数系列：

fsn='dh'

nums='1 2 3'

则它可以产生三个小波函数db1、db2和db3。

(5) 新建一个 *.mat 文件或 *.m 文件。wavemngr函数需要一个文件参数(file argument)。对于一个有许多小波函数的小波函数系，必须要定义一个 *.m 文件，这个m文件的格式与具体的小波函数类型有关；对于只有一个小波函数的小波函数系列，必须要定义一个 *.mat 文件。

(6) 对于③、④类型小波函数，因为它不具有紧支性，所以需要定义其有效支撑长度。

定义有效支撑长度就是指明该小波函数的上界和下界。例如对于已有的小波函数，其有效支撑长度的定义如表 2-6。

表 2-6　小波函数系列(Family)

小波函数系列(Family)	下界(Lower Bound)	上界(Upper Bound)
Meyer	-8	8
Mexican_hat	-5	5
Morlet	-4	4

新建一个 *.mat 或 *.m 文件：

wavemngr 函数需要一个文件参数，这个参数也就是 *.mat 或 *.m 文件的文件名。如果一个小波函数系中含有许多个小波函数，那么必须定义一个 *.m 文件，并且对于不同的小波函数类型，该文件具有不同的类型。如果一个小波函数系中只含有单个小波函数，那么对于第一类小波函数必须定义一个 *.mat 文件。在这个 *.mat 文件中，需要有该小波函数系的缩写名参数(fsn)，而且这个缩写名还要有一个数值参数(参数值代表的是尺度滤波器)。

第一类小波(正交的 FIR 滤波器)：

*.m 文件的第一行的语法结构是：function w = file(wname)

输入参数 wname 是一个小波函数名，输出参数 w 是相应的尺度滤波器，且 w 必须是偶数长度，否则 MATLAB 工具箱自动补 0，变成偶数长度。

在 MATLAB 工具箱中提供的小波函数系中，尺度滤波器的数值总和为 1，对于一个新的小波函数(这个新的小波函数尺度滤波器的数值不能为零)，可以不需进行正规化，因为 MATLAB 工具箱会自动对尺度滤波器进行正规化运算，使它的总和为 1。

以下是在 MATLAB 中已经定义的 *.m 文件：

对于 Daubechies 函数有 dbwavf.m 文件；

对于 Coiflets 函数有 coifwavf.m 文件；

对于 Symlets 函数有 symwavf.m 文件。

第二类小波(双正交的 FIR 滤波器)：

*.m 文件的第一行的语法结构是：function [wr,wd] = file(wname)

输入参数 wname 是一个小波函数名，输出参数 wr,wd 是相应的重构和分解尺度滤波，且 wr 和 wd 必须是偶数长度，通常最初的双正交滤波器不满足这个条件，MATLAB 工具箱自动补 0。

对于 MATLAB 工具箱中已经定义的小波函数，其尺度滤波器的数值总和为 1，对于一个新的小波函数，数值的总和可以是除零之外的任意值，由 MATLAB 工具箱自动将它进行正规化处理。

在 MATLAB 中已经定义的 *.m 文件有：

对于 BiorSplines 函数有 biorwavf.m 文件。

第三类小波(正交滤波器，具有尺度函数)：

*.m 文件的第一行的语法结构是：

function [phi,psi,t] = file(lb,ub,n)

输出参数 phi,psi 是相应的支撑长度为[lb,ub]的尺度函数和小波函数。

在 MATLAB 中已经定义的 *.m 文件有：

对于 Meyer 函数有 meyer.m 文件。

第四类小波(无 FIR 滤波器，无尺度函数)：

*.m 文件的第一行的语法结构是：

function [psi,t] = file(lb,ub,n)

输出参数 psi 是相应的支撑长度的上下界为[lb,ub]，且在长度为 t 的时间段内有 n 个点的小波函数。

在 MATLAB 中已经定义的 *.m 文件有：

对于 Mexican_hat 函数有 mexihat.m 文件；

对于 Morlet 函数有 morlet.m 文件。

2.8.2 如何添加一个新的小波函数系列

要添加一个小波函数系列，需要用到 wavemngr 函数，它有下面两种形式。

(1) wavemngr('add',fn,fsn,wt,nums,file)

(2) wavemngr('add',fn,fsn,wt,nums,file,b)

以下是对该命令函数用法的举例：

① wavemngr('add','Ndaubechies','ndb',1,'1 2 3 4 5','dbwavf');

② wavemngr('add','Ndaubechies','ndb',1,'1 2 3 4 5 * *','dbwavf');

③ wavemngr('add','Nbiorwavf','nbio',2,'1.1 1.3','biorwavf');

④ wavemngr('add','Nmeyer','nmey',3,'','meyer',[-8,8]);

⑤ wavemngr('add','Nmorlet','nmor',4,'','morlet',[-4,4]);

例 1：现在我们向系统中加入一个小波函数系列，该小波函数系列是由 Strang 和 Nguyen 提出的。

小波函数系列的全称名：Binlets

小波函数系列的缩写名：binl

小波类型：2(具有 FIR 滤波器的双正交小波)

小波函数系列中的序号有：7.9(这里只举出一例进行说明)

新建的 *.m 文件用于产生该小波函数滤波器，文件名为：binlwavf.m

解：要添加该小波函数系，可按下面的程序进行：

```
%程序清单
%添加一个新的双正交小波函数系列
wavemngr('add','Binlets','binl',2,'7.9','binlwavf')
%列举出所有的小波函数系列
wavemngr('read')
%输出结果：
    ans =
```

```
==============================
Haar              haar
Daubechies        db
BiorSplines       bior
Coiflets          coif
Symlets           sym
Morlet            morl
Mexican_hat       mexh
Meyer             meyr
Binlets           binl
==============================
```

如果你要获得该小波函数系的在线帮助信息，可以建立如下的相关帮助文件。

```
Function binlinfo
%BINLINFO 双正交小波函数系的信息(binlets)
%
%        双正交小波函数系(Binlets)
%        函数系列全称名                Binlets
%        函数系列缩写名                 binl
%        Nr, Nd 的顺序               Nr=7,Nd=9
%        正交性              no
%        双正交性              yes
%        紧支性        yes
%        离散小波变换                          possible
%        连续小波变换                          possible
%
%        binl Nr.Nd          ld effcetive        lr effcetive
%                            length of LoF_D     length of HiF_D
%        binl7.9                    7                    9
```

与产生滤波器相关的 *.m 文件是:

```
function [Rf,Df]=hinlwavf(wname)
%BINLINFO 双正交小波函数系的信息(binlets)
%    [RF,DF]=BINLWAVF(W) 返回两个尺度滤波器，其中
%    双正交小波函数由字符串 W 指明
%    W='binlNr.Nd' 其中 Nr 和 Nd 的值可取为如下:
%    Nr=7     Nd=9
%    输出参数是滤波器
%    RF 是重建滤波器
%    DF 是分解滤波器
%检验参数
```

```
if errargn('binlwavf',nargin,[0,1],nargout,[0:2]),error('*');
end
% 小波函数的扩展语句
Nr=7;Nd=9
% 以下是在 Binlets 函数系列中新扩展的小波函数
% ------------------------
if nargin==0
    Nr=7;Nd=9;
elseif isempty(wname)
    Nr=7;Nd=9
else
    if isstr(wname)
        lw=length(wname);
        ab=abs(wname);
        ind=find(ab==46 | 47<ab | ab<58);
        li=length(ind);
        err=0;
        if li==0
            err=1
        elseif ind(1)~=ind(li)-li+1
            err=1;
        end
        if err==0
            wname=str2num(wname(ind));
            if isempty(wname0,err=1;end
        end
    end
    if err==0
        Nr=fix(wname);Nd=10*(wname-Nr);
    else
        Nr=0;Nd=0;
    end
end
% 以下是函数扩展语句
% 加入出错测试
% ------------------------
if Nr~=7,Nr=7;end
if Nd~=9,Nd=9;end
if Nr==7
```

```
    if Nd==9
        Rf=[-1 0 9 16 9 0 -1]/32;
        Df=[1 0 -8 16 46 -8 0 1]/64;
    end
end
```

例 2：在这个例子中，要加入一个新的紧支的双正交小波函数系列到 MATLAB 工具箱中，该小波函数系列是 Dasubeches 小波函数系列一个派生小波函数系列，它的产生基于 Bernstein 多项式。

解：要在 MATLAB 环境中加入该小波函数，可按如下程序进行：

```
%程序清单：
%将最初的小波函数系列全部罗列出来
  wavemngr('read')
  ans=
  ===================================
  Haar              haar
  Daubechies        db
  BiorSplines       bior
  Coiflets          coif
  Symlets           sym
  Morlet            morl
  Mexican_hat       mexh
  Meyer             meyr
  ===================================
  %罗列出小波函数系列中的小波函数
  wavemngr('read',1)
  ans =
  ===================================
  Haar              haar
  ===================================
  Daubechies        db
  --  -------------------------
  db1     db2     db3     db4
  db5     db6     db7     db8
  db9     db10 db * *
  ===================================
  BiorSplines       bior
  ----------  -----------
  bior1.1     bior1.3     bior1.5     bior2.2
  bior2.4     bior2.6     bior2.8     bior3.1
```

```
bior3.3     bior3.5     bior3.7     bior3.9
bior4.4     bior5.5     bior6.8
=============================
Coiflets              coif
-------------------------
coif1   coif2   coif3   coif4
coif5
=============================
Symlets               sym
-------------------------
sym2 sym3 sym4 sym5
sym6 sym7 sym8
=============================
Morlet                morl
=============================
Mexican_hat           mexh
=============================
Meyer                 meyr
=============================
%加入新的正交小波函数系列，必须定义以下内容：
%    小波函数系列全称名                  Lemarie
%    小波函数系列缩写名                  lem
%    小波函数的类型                      1(正交)
%    小波函数系列的序号                  1 2 3 4 5
%    文件名                              lemwavf
%命令函数 lemwavf.m 必须是以下的形式：
%    function w=lemwavf(wname)
%    输入参数 wname 是一个字符串
%    wname='lem1'或'lem2',…,等等
%    wname=sh.name+number
%    w 是相应的尺度滤波器
    wavemngr('add','Lemarie','lem',1,'1 2 3 4 5','lemavf');
%    二进制文件'wavelets.asc'被另存为文件'wavelets.prv'，并且被修改；
%    产生了 mat 文件'wavelets.inf'
%    再列出小波函数系列中的各个小波函数系列
    wavemngr('read')
ans=
=============================
Haar                  haar
```

Daubechies	db
BiorSplines	bior
Coiflets	coif
Symlets	sym
Morlet	morl
Mexican hat	mexh
Meyer	meyr
Lemarie	lem

================================

2.8.3 添加小波函数系列之后

当用 wavemngr 函数添加一个小波函数系列后，工具箱会自动在当前目录生成三个文件：两个 ASCII 码文件 wavelets.asc 和 wavelets.prv 及一个 *.mat 文件 wavelets.inf。

如果用户想在小波工具箱中使用自己的小波函数系列，可以按如下步骤进行：

① 创建一个新的目录，以存放自己的小波函数系列；

② 把上面提到的文件移动到该目录；

③ 将这个新建的目录放入 MATLAB 的搜索路径中去(具体方法请参见 path 命令函数)；

④ 需要修改小波函数系列，最好在同一个目录中进行。每次都将生成的小波函数系列放在不同的目录中，可能会产生不可预料的结果；

⑤ 定义一个名为"<fsn>info.m"的 *.m 文件，例如可以是 dbinfo.m 或 morlinfo.m。

完成以上步骤之后，当用户在小波工具箱中的图形用户接口中单击 Wavelet Family 按钮时，这个文件就自动出现在小波显示选项框中，显示出这个新建的小波函数系列。

2.9 GUI 用法简介

小波函数工具箱提供的图形用户界面 GUI(Graphical User Interface)，它使 MATLAB 的使用变得更加简单、直观，可以在图形方式下实现小波分析的绝大部分功能。它不需编程，只要通过菜单操作和各种参数的选择就可对一维或二维信号进行小波分析。下面对这一工具作简单介绍。

2.9.1 概述

1. 进入 GUI 工作环境

启动 MATLAB 后，在出现的命令行窗口(Command Window)中键入 wavemenu 后回车，就可以进入小波工具箱主菜单窗口(Wavelet Toolbox Main menu)。MATLAB 提供各种不同的小波分析图形工具，如图 2.97 所示。

小波分析的 GUI 提供的图形工具有小波和小波包信息显示工具、一维连续小波分析、一维和二维离散小波分析、一维和二维离散小波包分析工具。

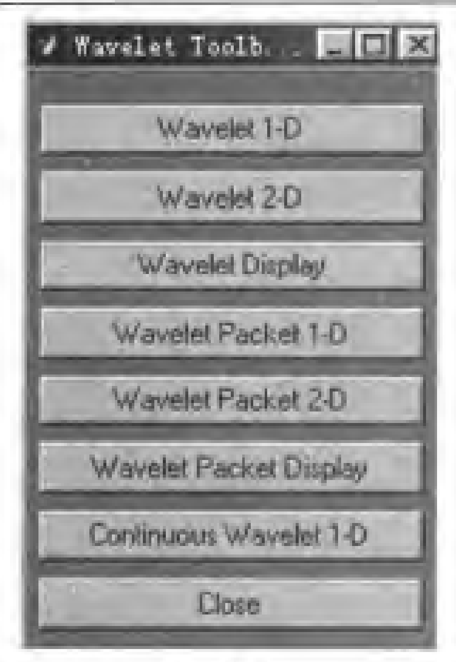


图 2.97

2. GUI的主要特征简介

它有以下一些主要特征：

· 小波函数的选择。在窗口的右上角，可以选择小波函数和小波分解的层数。图形工具提供的小波函数只有五种，即haar，db，bior，sym和coif，用户也可以自己添加小波函数。

· 颜色设置。在GUI工作环境中，它对原始信号以红色显示，重构或合成的信号以黄色显示，低频信号以蓝色显示(层数越高，颜色越深)，高频信号以绿色显示(层数越高，颜色越深)。

· 曲线绘制的关连性。相关信息的数据在绘制曲线时，横坐标是同一个尺度基准，当一条曲线的横坐标变化时，其它的横坐标也随之变化，称为曲线绘制的关连性。例如，对某一信号的原始曲线局部放大，与之对应的高频和低频曲线也都以相同的尺度被局部放大。

· 对鼠标的使用。可用不同类型的鼠标对小波函数工具箱的GUI进行控制。

对二键鼠标：选择或激活控制(单击左键)、显示十字形或与位置相关的信息(单击右键)、调整曲线的上下和左右位置(左右键同时单击)。

对三键鼠标：选择或激活控制(单击左键)、显示十字形或与位置相关的信息(单击中键)、调整曲线的上下和左右位置(单击右键)。

· 控制颜色映射。在窗口的右下角，有一个Colormap选择框，它用于调节颜色映射，使二维图像的显示或一维曲线的绘制以相应的颜色映射显示出来。

· 控制颜色的数目。滑动条Nb.Colors位于窗口的右下角，它在二维图像的显示或一维曲线的绘制时，调节颜色的数目。当用鼠标拖动调节时，其右侧的文本框中的数字相应改变。也可以直接在文本框中输入数字，调节颜色数目。正确地选择颜色数目，可以突出地显示曲线的某些特征。

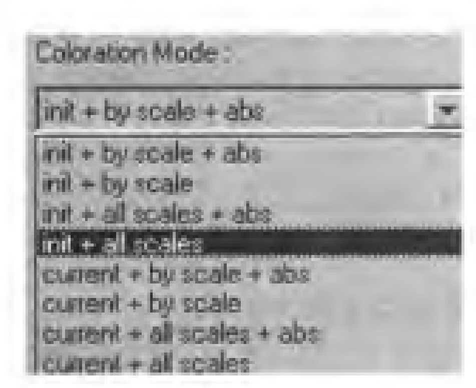


图 2.98

· 控制染色模式。染色实际上就是将大小不同的数据以多种不同的颜色显示出来，可以通过不同的颜色来表征数据的大小，这在一维连续小波工具、二维离散小波分析工具，以及一维和二维小波包分析工具中很有用。在连续小波工具中，窗口的右边有一个Coloration mode(染色模式)选择框，可以选择几种不同系数的染色。在一维小波工具中，当在窗口右边Display mode(显示模式)选择框中选择Show and scroll、Separate mode和Superimpose mode时，其下方的More display options按钮呈激活状态，单击(在本节中，为方便起见，“单击”表示“鼠标左键单击”)该按钮，则可以显示Coloration mode选择框，该选择框的内容如图2.98所示。

染色系数有三个参数进行设置：

① 当选择 init 时，对所有的系数都进行染色，即将这些数据分成若干等分（等分的数量与颜色数目 Nb. Colors 相同），然后将所有数据以对应的颜色显示出来。

② 当选择 current 时，只染色当前系数，即只对当前选中的系数数据按照颜色数目进行等分，这时显示的颜色只反映当前层的系数大小，而不能反映其它层的系数大小。在选择 current 后，可以通过 Apporiximates Axes 和 Detail Axes 下方的标记框（Checkbox）来选择需要显示的系数。

by level 或 all levels（只在一维离散小波工具中适用）、by scale 或 all scales（只在一维连续小波工具中适用）：当选择 by level 或 by scale 时，单独对每一个高频层或高频尺度进行染色；否则，对所有分解层或分解尺度都染色。

③ 当选择 abs 时，在系数染色之前，先将系数取绝对值。

· 设制自己的图形对象：用户可以根据自己的习惯，将窗口中的各种图像和曲线按照自己的风格显示。在窗口中选择 Option→Handles Graphics Setting 菜单选择，就可以看到几个子菜单。

当选择子菜单 Axes Settings 时，系统会弹出如图 2.99 所示的窗口，问你是否需要动态显示工具（Dynamical Visualization Tool，缩写为 DVT）。

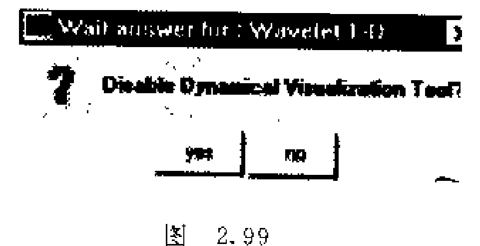


图　2.99

在完成设制图形之后，必须将 Axes Setting 子菜单失效（Disable），以重新激活 DVT。

当选择子菜单 Window Setting 有效（Enable）时，你可编辑当前的图形参数。

当选择子菜单 Axes Setting 有效（Enable）时，在一个坐标轴上单击鼠标左键，可编辑与坐标有关的各种参数。

当选择子菜单 Text Setting 有效（Enable）时，在一个坐标轴上双击鼠标左键，可编辑与坐标有关的各种参数。

· 演示程序的使用。在小波函数图形工具中，每一个分析工具都有相应的演示程序，用户可以在 File 菜单中的 Demo Analysis 子菜单中调用这些演示程序，它可以让用户选择一个分析实例，这些分析实例是预先定义参数的。当选择 File→Demo Analysis 菜单选项时，有一个窗口弹出，它列出了所有的例子，其外形如图 2.100 所示。

```
with db3 at level 5  --->  Sum of sines
with db5 at level 5  --->  Frequency breakdown
with db3 at level 5  --->  Uniform white noise
with db3 at level 5  --->  AR(3) noise
with db2 at level 4  --->  Noisy polynomial
with db3 at level 4  --->  Noisy polynomial
with db2 at level 5  --->  Step signal
with db2 at level 5  --->  Two nearby discontinuities
with db7 at level 5  --->  Two nearby discontinuities
with db1 at level 2  --->  Second derivative breakdown
with db4 at level 2  --->  Second derivative breakdown
with db3 at level 6  --->  Ramp + white noise
with db3 at level 6  --->  Ramp + colored noise
with db5 at level 5  --->  Sine + white noise
with db5 at level 6  --->  Triangle + sine
with db5 at level 7  --->  Triangle + sine + noise
```

图　2.100

选择其中任何一个例子，就进入了它的演示程序，每个例子均有自动演示功能，可以自动观看它的分析过程，也有单步演示功能，可以根据用户的需要，一步步地进行演示。这些小波分析的演示程序对于学习 GUI 的使用方法很有帮助，由于它是一个单纯的菜

单操作，并且每一步都有提示信息，使用起来十分方便，故对此不再赘述。

3. GUI工作环境中的数据I/O

（1）一维小波（或小波包）分析中的数据I/O。

在一维小波GUI环境下，用户在对信号进行小波分析的过程中，将各种分析数据存储为一个*.mat文件。

在File菜单下，有六个与数据I/O有关的子菜单，它们是Load Signal（装入信号）、Load Coefficients（装入系数）、Load Decomposition（装入分解后的信号）、Save Synthesized Signal（保存合成后的信号）、Save Coefficients（保存系数）和Save Decomposition（保存分解后的信号）。这六个子菜单在一维小波分析图形工具箱窗口中的位置如图2.101所示。



图 2.101

在图形工具箱中，可以用File→Save Coefficients菜单选项来保存系数。保存时，会弹出一个文件存盘对话框，可以输入相应的文件名进行系数数据保存，如果不输入后缀名.mat，则系统会自动为文件名加上该后缀名。也可以用File→Save Synthesized Signal菜单选项来保存小波合成后的信号，其操作与保存系数时相同。还可以用File→Save Decomposition菜单选项来保存一维小波分解后的信号数据，这时，该数据文件中包含信号小波分析后的所有分析数据，它默认的文件后缀名是.wa1（wavelet analysis 1－D）。

以后缀名.wa1存盘的文件也是mat格式的文件，但它在文件的开头部分加入了一个向量，用来指示在MATLAB数组（在MATLAB环境中，将所有数据，例如向量和矩阵都看作是MATLAB数组）的数据中各个小波分解层次的数据在该数组文件中的大小及位置。

在File菜单下，也可以随时装入待分析的信号。通过File→Load Signal菜单选项或File→Load Coefficients菜单选项，装入以.mat为后缀名的信号数据或小波分解系数数据。通过File→Load Decomposition菜单选项，装入以.wa1为后缀名的小波分解后的信号数据。

（2）二维小波（或小波包）分析中的数据I/O。

在二维小波函数GUI环境下，用户也可以在对二维信号进行小波分析的过程中，进行数据的存储，二维小波图形工具箱可以将数据存成mat格式的文件。

在File菜单下，同样有六个与数据I/O有关的子菜单，它们是装入图像、装入系数、装入分解后的图像、存储合成后的图像、存储系数和存储分解后的图像。这六个子菜单在二维小波分析图形工具箱窗口中的位置如图2.102所示。

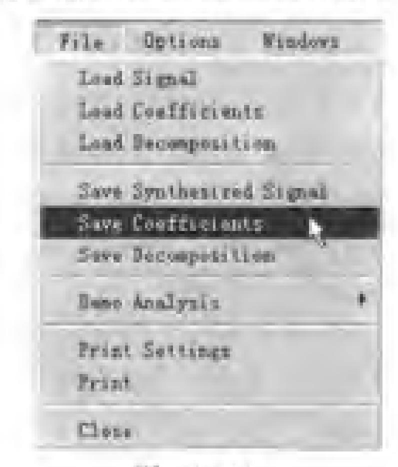


图 2.102

可以用File→Save Coefficients菜单选项来存储系数。在图形工具箱中，存储时会弹出一个文件存盘对话框，可以输入相应的文件名进

行系数数据存储，如果不输入后缀名.mat，则系统会自动为文件名加上该后缀名。也可以用 File→Save Synthesized Image 菜单选项来存储小波合成后的图像，其操作与存储系数时相同。还可以用 File→Save Decomposition 菜单选项来存储二维小波分解后的图像数据，这时，该数据文件中包含图像小波分析后的所有分析数据，它默认的文件后缀名是.wa2（wavelet analysis 2—D）。该文件中包含的是二维图像小波分析后所有的数据，包含原始图像、小波分解系数数据、小波重构数据。

在 File 菜单下，也可以随时装入待分析的图像。通过 File→Load Image 菜单选项和 File→Load Coefficients 菜单选项，装入以 *.mat 为后缀名的图像数据及小波分解系数数据。通过 File→Load Decomposition 菜单选项，装入以.wa2 为后缀名的二维小波分解后的图像数据。

由于在 GUI 环境下的数据存储操作简单，并且数据存储的格式和在命令行环境下的数据存储格式完全一样，所以，在此不再举例对其用法进行说明。但需要强调的一点是，在小波 GUI 环境下，若想将一个 *.mat 文件调入 MATLAB 工作环境中去，则在调入数据之前，需要预先知道所调入的文件是一维数据还是二维数据。因为在 MATLAB 中，一维数据和二维数据都是以 *.mat 文件存储的，这些文件的后缀名虽然是一样的，但是内部的数据结构是不一样的，而在小波 GUI 环境下，它在调入数据时，不能区分一维数据和二维数据。如果在一维图形分析工具中装入的是一个二维数据，或者在二维图形分析工具中装入的是一个一维数据，则会产生出错信息。

2.9.2　一维连续小波图形工具简介

在这里，我们以一个电网监测的电压曲线的分析实例来介绍该工具的用法。

(1) 在小波函数工具箱主菜单窗口中用鼠标单击 Continuous Wavelet 1—D 按钮，就可以进入一维连续小波图形分析工具环境。选择菜单 File→Load signal，在弹出的 Load Signal 对话框中选择待分析的信号文件名并单击“打开”按钮，则在 analyzed signal 图形框中显示出该信号的波形。此例中选择的是一个电压信号 leleccum.mat，这时，窗口如图 2.103 所示。

在 Scale Values 选择框中，可以选择单步模式(Step by step Mode)、2 的指数模式(Power 2 Mode)和手动模式(Manual Mode)这三种尺度模式。每种尺度模式都有自己的尺度特征。

· 在单步模式下，用户可以在文本框中输入最小的尺度值(min)、最大的尺度值(max)和尺度的步长(Step)。

· 在 2 的指数尺度模式下，小波分解的尺度是 $2^0, 2^1, 2^2, \cdots, 2^k$，其中 k 的大小由 Power 选择框来选择。在这种尺度下，其实和二进制离散小波是一样的。

· 在手动尺度模式下，用户可以在文本框中按照 MATLAB 的语法和自己的要求，输入尺度向量。用户也可以以任意的尺度对信号进行小波分解。

现在，我们在窗口的右边选择 haar 小波，在 Scale values 选择框中选择单步尺度(Step by step Mode)，在最小尺度文本框(min)中输入 1，最大尺度文本框(max)中输入 30，步长文本框(Step)中输入 1，Nb. Corlors 文本框中输入 256，其它设置为默认值，单击 Analyze 按钮，则显示分解后的图形，如图 2.104 所示。

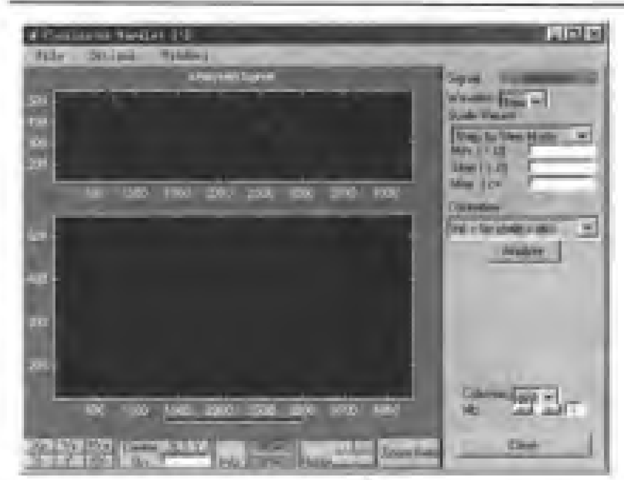


图 2.103

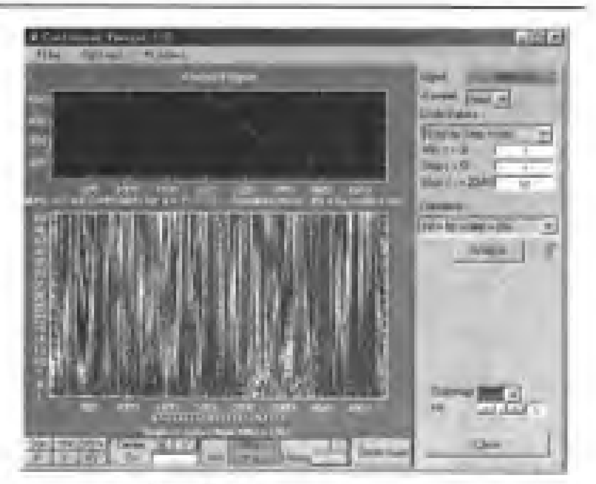


图 2.104

在MATLAB中，可以对一维或二维信号进行局部放大分析。用户若只对信号的局部感兴趣，也可以利用MATLAB的局部放大功能进行放大，它的使用也很简单。现在，我们对这个信号进行局部放大分析：按下鼠标左键，并拖动鼠标，会显示一个红色的方框，外形如图2.105所示。



图 2.105

选中放大的区域后，松开鼠标左键，然后在窗口的底部点击相应的放大方式。在这里，我们只对水平轴进行放大，即在Zoom栏中，用鼠标点击x+按钮，则会立即显示放大后的信号及相应的小波分解结果，如图2.106所示。

在对信号的局部进行小波分析后，可以将分析数据存盘，选择菜单中的File→Save Coefficients即可。

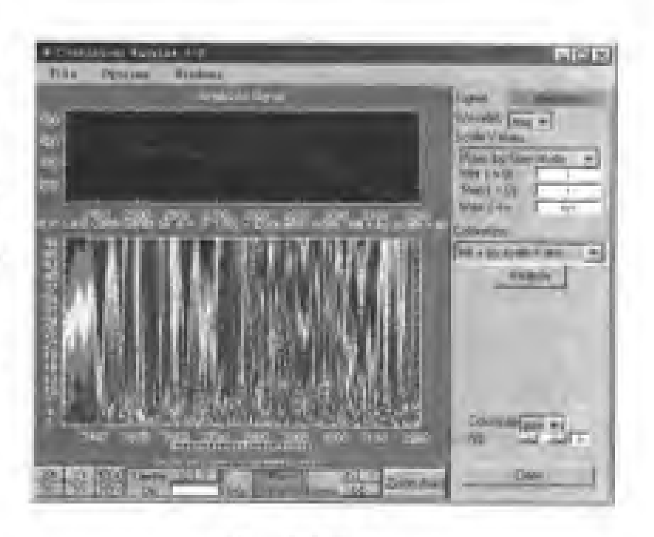


图 2.106

2.9.3 一维离散小波图形工具简介

在这里，我们同样以一个电网监测的电压曲线的分析实例来介绍该工具的用法。在Load Signal对话框中选择leleccum.mat信号并单击open按钮。在Wavelet和level选择框中分别选用db3和5并单击Analyze按钮，即可以对信号进行五层小波分解。分解后窗口的显示如图2.107所示。在默认情况下，一个信号用小波分解后，它的显示模式是完全分解模式(Full Decomposition)。

为了方便用户分析信号，MATLAB提供了几种信号的显示模式。这几种显示模式可以在显示模式选择框(Display mode)中进行选择。显示模式选择框的外形如图2.108所示，它有显示滚动模式(Show and Scroll)、完全分解模式(Full Decomposition)、分离模式(Separate Mode)、叠加模式(Superimpose Mode)和树模式(Tree Mode)。

(1) 显示滚动模式。在这种模式下，无论将信号分解到第几层，每次只分别显示一个低频系数和一个高频系数，具体显示第几层的系数，则由App.和Det.选择框确定。系数的显示设置可单击More Display Options按钮，在More Display Options弹出窗口中设置参数来实现，其窗口显示如图2.109所示。

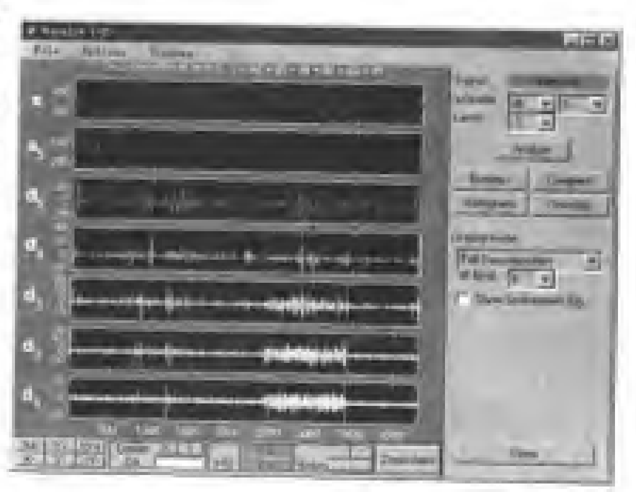


图 2.107

在该窗口中的标记框中，可以选择需要显示的内容，在选择完毕后，单击Apply按钮，就可看到系数已按照自己选择的风格显示出来了。各个标记框中的具体特性是：

· Show Appro. Axes用于控制原始信号及低频分解系数的重构信号的显示。选中这个标记框后，会出现Show Signal和Show Synth. Signal两个子标记框，Show Signal是显示原始信号；Show Synth. Signal是显示某层小波分解低频系数的重构信号。



图 2.108

· Show Detial Axes用于控制原始信号及高频分解系数的重构信号的显示。选中这个标记框后，会出现Show Signal和Show Synth. Signal两个子标记框，Show Signal是显示原始信号；Show Synth. Signal是显示某层小波分解高频系数的重构信号。

· Show Coefs. Axes用于显示分解高频系数，它通过颜色映射(Colormap)、颜色数量(Nb. of color)以及染色模式(Coloration Mode)来控制显示方式，以灰度图像的形式显示出各层的系数大小。

该模式下，分解后信号的显示如图2.110所示。



图 2.109

(2) 完全分解模式。它将信号分解的高频部分和低频部分可以全部显示出来，这是MATLAB图形分析工具一种默认的显示方式。在该方式下，具体显示到第几层，由at level选择框决定。选中Show Synthesized Sig标记框，可显示合成后的信号，合成后的信号与原始信号在同一个坐标系中显示。如分解五层后，将五层全部显示出来的窗口如图2.111所示。

(3) 分离模式。它和完全分解模式的显示类似，但它将高频部分和低频部分的系数分成两列来显示。具体显示第几层的系数，由单击More Display Options按钮后弹出的More Detail Options窗口来设置。在该设置窗口中，选择要显示的层数后，单击Apply按钮，就可看到系数已按照自己的风格显示出来了。在这里，我们对小波分解后低频系数只选中第1、2层，高频系数选中第1、2、3、4层，并选中显示高频系数的灰度图像，分析结果如图2.112所示。

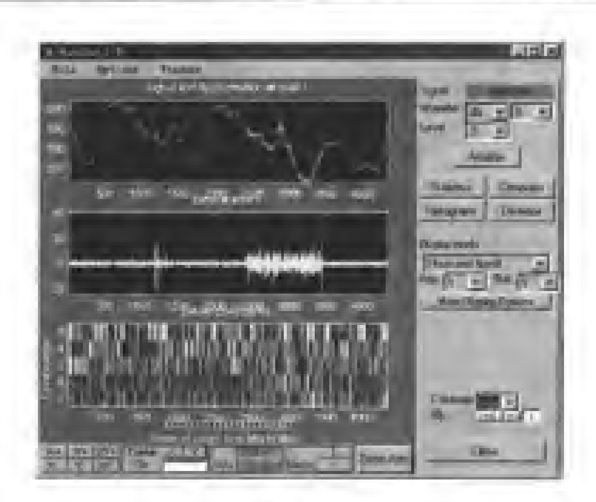


图 2.110



图 2.111

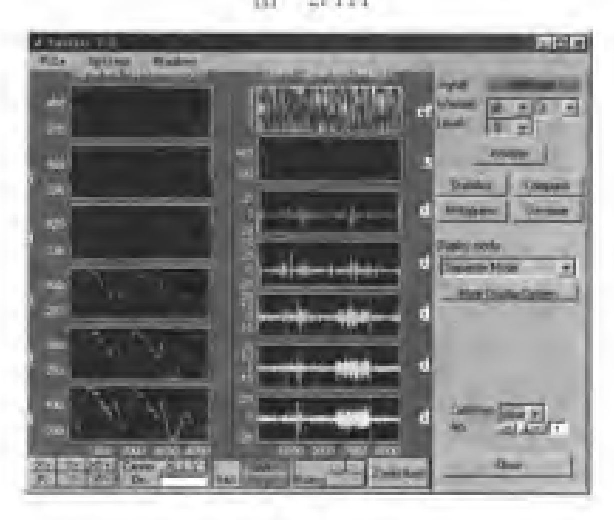


图 2.112

(4) 叠加模式。该模式下的显示结果与显示和滚动模式(Show and Scroll)基本相同，不同之处在于叠加模式是将低频的各层系数都叠加在一起，在一个方框中显示，将高频部分的各层系数都叠加在一起，也在一个方框中显示，各层系数用不同颜色的曲线表示。这时，

可以通过单击 More Display Options 按钮后，选择弹出的窗口中的各种参数来决定显示的具体内容。如果选中所有层的低频和高频系数时，则其分析结果如图 2.113 所示。

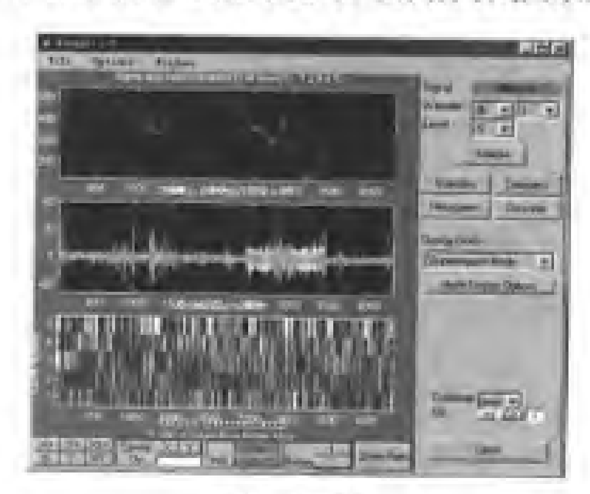


图 2.113

(5) 树模式。在该模式下，显示区域内有三个方框，最左边的方框用于显示小波分解的树结构；右上角的方框用于显示原始信号；右下角的方框显示的内容由用户自己选择，具体为：用鼠标单击小波分解树中的任一个结点(s 或 a_i 或 d_i)，则在这一方框中显示该结点系数的重构信号。在这里，我们点击 d_5 时，分析结果如图 2.114 所示。

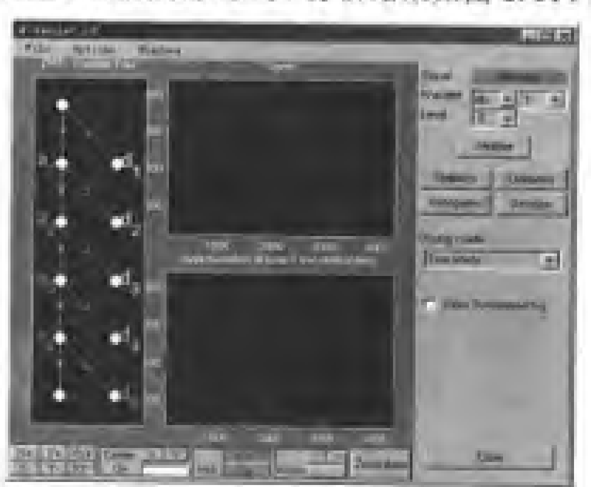


图 2.114

注意：在这几种显示模式下，如果选中 Show Synthsized Sig. 标记框，则合成后的信号会在原始信号的显示方框中以黄色曲线显示出来。

在信号用小波分解后，窗口的右边有四个按钮，它们是 Statistics(统计)、Compress（压缩）、Histograms（直方图)和 De-noise（消噪)，这四个按钮的外形如图 2.115 所示。

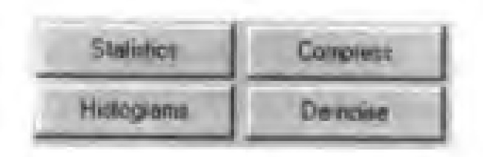


图 2.115

① 统计按钮执行的操作是对信号的分布情况进行统计。它在计算出信号序列 $f(n)(n=0,1,2,\cdots,N-1$，其中 N 是信号序列的长度)的平均值、最大值、最小值、中值、标准偏差等表征统计特性的值后，以文本方式显示在窗口左下方，同时，它还显示该信号的直方图和求和

形式的直方图(Cumulative Histogram)。

一维信号的直方图：

对一维函数 $f(x)$ 的采样序列 $f(n)$，$(n=0,1,2,\cdots,N-1)$，则表示它的幅值分布密度函数 $p_k=n_k/n$ 的关系图就称为一维信号的直方图。其中 k 是采样序列 $f(n)$ 的幅值等级，n_k 是在第 k 个幅值等级中的点列个数，$(k=0,1,2,\cdots,L-1)$，N 是总的点列个数。在MATLAB的一维离散小波图形工具中，其幅值的单位等级为1，即若信号的幅值为－200～300，则幅值等级分为200＋300＝500级，即L＝500。由直方图可以看出信号幅值大小的分布情况，若 $f(n)$ 是一个频域内的函数，则它可以表述信号频率的分布情况。

设 $p(k)$ 为信号的直方图函数，则求和形式的直方图是：

$cp(k) = p(k-1) + p(k)$，$k = 1,2,\cdots,L-1$（其中，L 是信号的幅值等级）

单击该 Statistics 按钮后，在弹出的窗口右边有四个单选钮(这四个选择钮中，每次只能选择其中的一个)，分别是 Orignal signal、Synthesized signal、Approximations 和 Details，选择其中某个单选钮，则在左边的显示区域中可以显示相应信号的统计结果，包括文本方式显示的计算值、以图形方式显示的直方图和求和形式的直方图。当 Approximations 和 Details 被选中后，在窗口的中部会出现 Approximattions Level 和 Details level 的选择框，让用户具体选择要显示的低频系数或高频系数的层次。柱条数(Number of bins)文本框可以让用户输入两种直方图中显示的柱条数量。在各项设置完成后，单击 Show Statistics 按钮，则经过计算后，在窗口的左边显示出统计结果。注意，每次只能选择一种信号进行统计。在这里，我们将柱条数选为30，选择第五层低频系数，对小波分解后的数学特征进行统计，统计结果如图2.116所示。

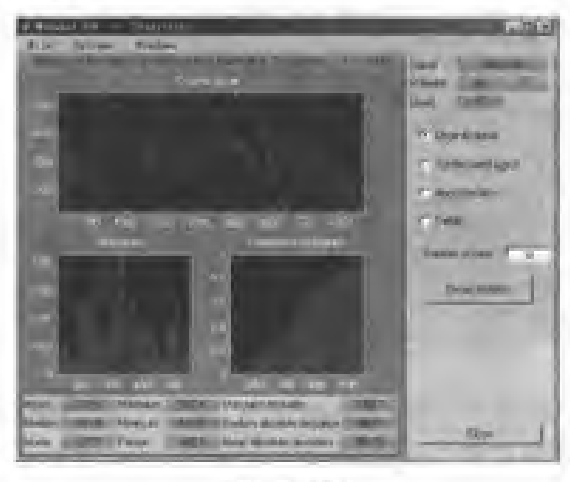


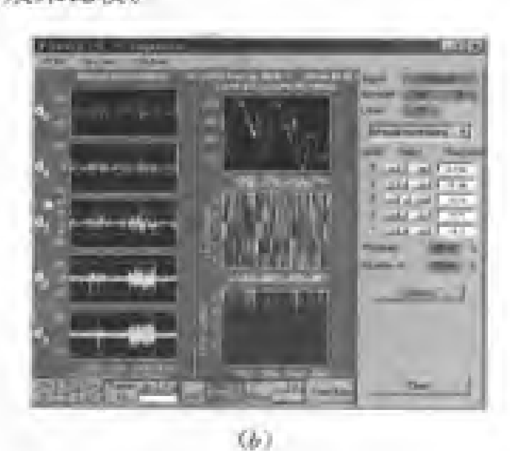
图　2.116

(2) 压缩按钮执行的操作是将信号数据用小波方法进行数据压缩。单击该按钮后，弹出数据压缩窗口。它有一个阈值选择框，可以选择自动阈值(Automatic thresholding)或手动阈值(Mannual thresholding)进行小波分解的系数阈值量化。当进行自动系数阈值量化时，该阈值是一个全局阈值，可在 Select global threshold 选择框中拖动滚动条或在其文本框中输入数值来选择该阈值，也可在左边的 Automatic thresholding 显示方框中用鼠标拖动绿色的阈值虚线来选择。当用鼠标拖动阈值虚线左右移动时，Select global threshold 选择框中的滚动条和其右边文本框中的数值也相应改变。当进行手动系数阈值量化时，用户可以

对高频系数的每一层都选择不同的阈值，每层阈值的选择方法和自动系数阈值量化中的全局阈值选择方法类似，不再详述。在阈值选择完成后，单击 Compress 按钮，则显示出压缩后的信号。压缩后的信号与原始信号相比，能量的百分比和置零系数个数的百分比也以文本方式在窗口中显示出来。在图 2.117 中，显示的是用自动阈值(图 2.117(*a*))和手动阈值(图 2.117(*b*))进行系数阈值量化后信号压缩的效果比较。



(*a*)



(*b*)

图 2.117

(3) 直方图按钮执行的操作是计算原始信号、重构信号以及分解的低频系数和各层高频系数的直方图并显示。

单击 Histogram 按钮后，在出现的窗口右边有四个标记框，分别是 Orignal signal、Synthesized signal、Approximations 和 Details，选择其中某个标记框，则在左边的显示区域中可以显示相应信号的直方图。Approximations 和 Details 选中后，在窗口的中部会再出现 Approximattions Level 和 Details level 的标记框，它可以让用户具体选择要显示的低频系数和高频系数的每个层次。在窗口的右下部，有两个选择钮，分别是系数(Coefficients)和重构(Reconstructed)，"系数"指直方图计算的是低频和高频部分的系数，"重构" 指直方图计算的是低频和高频部分的系数经过信号重构后，信号重构的幅值。文本框柱条数(Number of bins)，可以让用户输入直方图中显示的柱条数量。在各项设置完成后，单击 Show Histograms 按钮，经过计算后，会在窗口的左边显示相应直方图。在这里，我们将柱条数选为 30，对原始信号、重构信号和小波分解的全部高频和低频系数都进行直方图计算和显示，结果如图 2.118 所示。

(4) 消噪按钮执行的操作是对信号进行消噪处理。信号的消噪处理首先需要选择阈值的形式，在该图形工具中，有四种形式：自动硬阈值、自动软阈值、手动硬阈值和手动软阈值。在用手动阈值处理时，其操作与信号压缩中的手动阈值操作一样。可以在 Select Thresholding method 选择框中选择固定形式的阈值、硬 SURE 熵、递推 SURE 熵和极大极小熵中的任一种。由于在信号消噪中，需要对噪声的模型有先验知识，对噪声的类型要预先知道，所以在窗口的右下方还有一个 Select noise struct 选择框，可以让用户选择噪声的模型，这几种噪声模型是尺度未知的白噪声、尺度已知的白噪声和非白噪声。在这里，阈值的形式选择自动软阈值，递推 SURE 熵，噪声模型选择尺度未知的白噪声。去噪后的信号如图 2.119 所示。

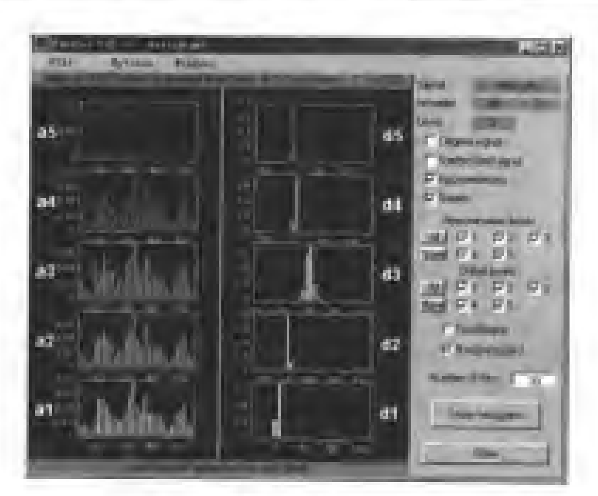


图 2.118



图 2.119

2.9.4 二维离散小波图形工具简介

利用二维离散小波图形工具可以方便地进行二维图像的小波分析。它的使用与一维小波图形工具类似，非常简便。下面我们在 wavelet 2－D 中调入 MATLAB\ Toolbox\ wavelet\ wavedemo 目录下的 woman.mat 图像文件，并以对它的分析实例来说明该工具的用法。

在窗口的右边选择 sym5 小波，分解层数选择 2，单击 Analyze 按钮，在左边显示区域中会显示出对 woman.mat 图像进行小波分析后的情况。在显示模式(View mode)中，可以选择小波分解后的图像显示方式，显示方式有两种。一种是方块图(Square)方式，它将分解后的图像以方块叠加的形式显示，另一种是树(Tree)模式，它将分解后的图像以树的层次形式显示。两种显示方式只是形式的不同，它们都能完全表征图像的信息。图 2.120 是方块图模式，图 2.121 是树模式。

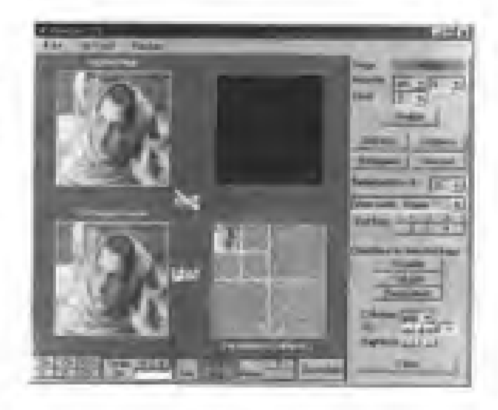


图 2.120

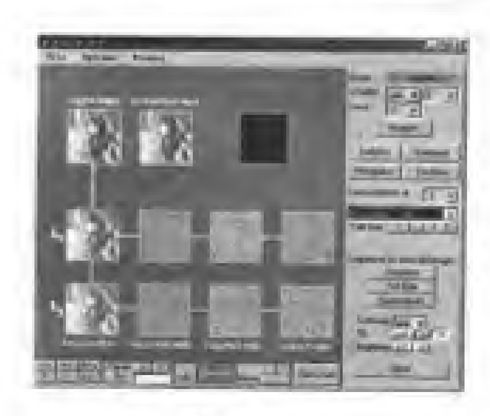


图　2.121

在窗口右边还有一个图像操作(Operations on selected image:)选择栏，它的下面有三个按钮，分别是：Visualize(显示)、Full size(全屏)和 Reconstruct(重构)。单击 Visualize 按钮，可以在空白图像框中显示该图像；单击 Full size 按钮，可以将该图像在窗口左边以全屏形式显示，这时 Full size 按钮变成 End full size 按钮，单击它可以返回先前的状态；单击 Reconstruct 按钮，可以将该图像系数进行重构，并显示重构后的图像。当用鼠标单击分解后的子图像时，该图像周围有一个绿色的边框，表示该图像被选中，可以在以上三个按钮操作方式中任选一种操作。

与一维小波图形工具一样，在窗口的右边也有 Colormap(颜色映射)、Nb. codor(颜色数量)和 Brightness(亮度)三种参数设置，这三种参数设置和一维的类似，故在此不再详述。

二维小波图形工具也能进行四种分析，即统计、压缩、直方图和消噪。它们分别由 Statistics(统计)、Compress (压缩)、Histograms (直方图)和 De-noise (消噪)四个按钮来执行，下面对这四种分析进行说明。

(1) 统计。MATLAB 只能对索引图像(Index Image)格式存储的二维图像进行分析。对一般的二维彩色图像，需要将它转化成索引图像，即单一颜色映射的图像格式，这种转化与图像的灰度转化类似，具体转化方法请见第 3 章。将图像进行索引图像的转化之后，图像矩阵中的每一个元素的值对应的是该点的颜色亮度，值越大，则该点的亮度越大，这是一个线性对应关系。Statistics 按钮执行的操作是对二维图像的亮度分布情况进行统计。对一个 $M\times N$ 的图像矩阵，在计算出图像矩阵每个像元亮度 $f(m,n)$(其中 $m=0,1,\cdots,M-1$；$n=0,1,\cdots,N-1$)的平均值、最大值、最小值、中值、标准偏差等表征统计特性的值之后，将其结果以文本方式显示在窗口左下方，同时，它还显示该图像矩阵的直方图和求和形式的直方图(Cumulative Histogram)。

亮度图像直方图的定义为

设亮度离散图像的亮度 $f(m,n)$ 共分为 L 级，即 $k=0,1,2,\cdots,L-1$，则它的像元值 $f(m,n)$ 处于第 k 层的亮度值的频率为 $p_r(r_k)=\frac{n_k}{n}$。式中 n 是图像中像元的总数，$n=M\times N$，n_k 是亮度处于第 k 级的像元数。当 k 从 0 到 $L-1$ 取值时，r_k 与 $p_r(r_k)$ 的关系图就称为图像亮度的直方图。

若 $p(k)$ 为图像矩阵的直方图函数，则求和形式的直方图是 $cp(k)=p(k-1)+p(k)$，其中 $k=0,1,2,\cdots,L-1$。

点击 Statistics 按钮后，弹出一个窗口，它的右边有四个单选钮(这四个选择钮中，每次

只能选择其中的一个)，分别是 Orignal image、Synthesized image、Approximations 和 Details。选择其中某个单选钮，则在左边的显示区域中可以显示原始图像、重构图像、图像分解的低频部分或高频部分的统计结果，包括计算统计量特征值、直方图和求和形式的直方图。当 Approximations 和 Details 选中后，在窗口的中部会出现 Approximattions Level 和 Details level 的选择框，用户可以选择要显示的低频系数或高频系数的层数。文本框柱条数(Number of bins)可以让用户输入两种直方图中显示的柱条数量。在各项设置完成后，单击 Show Statistics 按钮，则经过计算后，在窗口的左边显示出统计结果。注意，每次只能选择其中的一种图像进行统计。该分析原理与一维小波工具中的统计相同，操作方法上也大同小异，唯一不同的是：当选中 Details 单选钮后，会出现 Horizontal(水平)、Vertical(垂直)和 Diagonal(斜线)三种选择框，分别表示统计的是图像分解在水平、垂直和斜线方向上的统计量。在这里，我们选择第二层高频水平系数，柱条数设为 30，进行分析后，结果显示如图 2.122 所示。

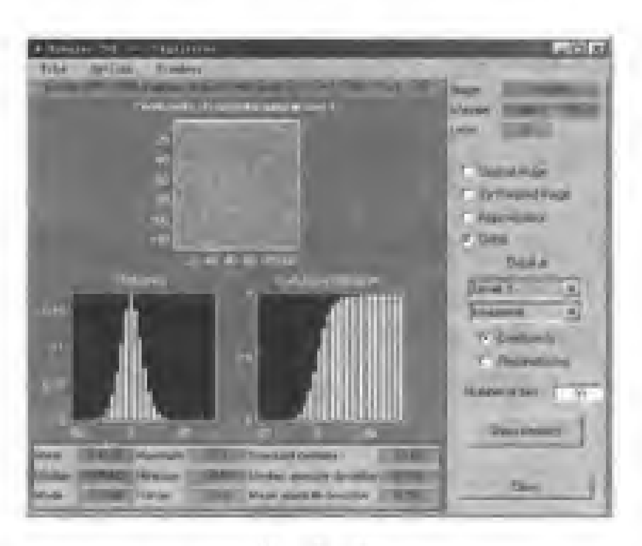


图 2.122

(2) 直方图。对图像进行分解后，计算并显示原始图像、重构图像、各层低频和高频部分的直方图。与 Statistics 情况类似，它只计算原始图像、重构图像、图像分解的低频部分和高频部分的直方图。其中，高频部分可分为水平方向上的高频部分、斜线方向上的高频部分和垂直方向上的高频部分。用户可以单击 Origional image 选择框后再单击 Show histograms 按钮来显示原始图像的直方图，也可以单击 Synthesized image 按钮显示重构图像的直方图。对于图像的多层分解，当用户选中 Approximations 选择框或 Details 选择框后，在它们的下方，会相应出现 Approximation levels 或 Detail levels 的二级选择框。这时，用户可以在这里选择图像分解后显示的层次，对高频还可以在水平部分、斜线部分和垂直部分这三个选项中任意选择一个来显示。在这种操作中，原始图像、重构图像、各层低频部分和高频部分可以同时显示在左边的显示区域内。在这里，我们对原始图像、重构图像、图像两层分解后的两个低频部分和两个高频水平部分一起显示在左边的显示区域内，这时的直方图显示窗口如图 2.123 所示。

(3) 压缩。将图像矩阵的像元数据用小波方法进行数据压缩。单击 Compress 按钮后，数据压缩显示窗口弹出。在这里它只能通过对图像小波分解后的高频系数进行阈值量化来

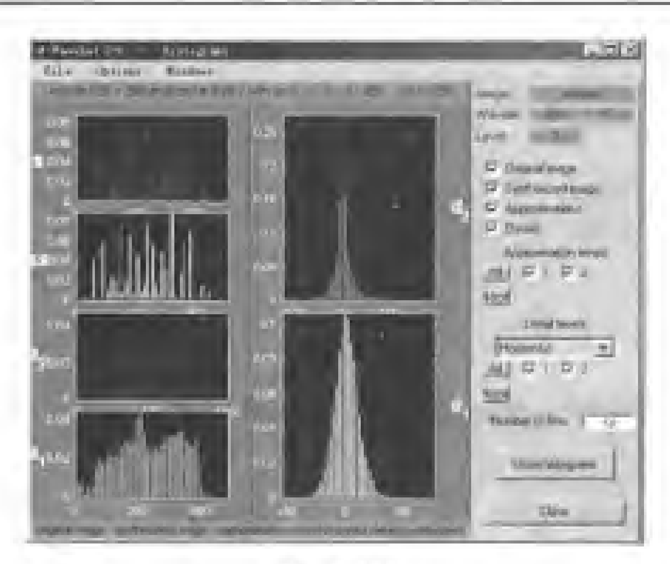


图 2.123

进行数据压缩。由于高频系数有水平方向、斜线方向和垂直方向三个部分，所以在这里也相应有三个选择框可供选择，用户可以分别给它们设置阈值之后，再单击Compress按钮即可进行图像数据的压缩。对于阈值选择方式有三种：① 可以在Select滚动条中，拖动滚动条来选择；② 可以在其右边的Threshold文本框中输入数值来选择；③ 可以在左边显示区域的显示方框中用鼠标拖动绿色的阈值虚线来选择。这三个部分的阈值选择方式是一样的。对一个图像进行多层小波分解，则对每一个分解层都可对三个高频分解系数进行阈值量化。这时，在窗口的右边，各层的阈值选择滚动条和文本框都按照层次大小，从上到下排列。在阈值选择完成后，单击Compress按钮，则显示出压缩后的图像，以及保留能量的百分比和置零系数个数的百分比。在这里，各层经过阈值量化后，图像压缩的结果如图2.124所示。



图 2.124

(4) 消噪。在图像的消噪中，阈值选择框中显示的是Mannual soft thresholding(手动软阈值)和Manual hard thresholding(手动硬阈值)。当一个含噪声的图像经小波分解后，噪声信息主要分布在高频的三个部分(水平方向、斜线方向和垂直方向)，所以图像的消噪也只针对高频的三个部分进行。消噪时，首先需要在水平方向、斜线方向和垂直方向中的每一

个部分选择相应的阈值大小，同时选择阈值的类型(即软阈值还是硬阈值)；在以上这些选择完成之后，单击De-noise按钮，经过分析计算之后，在窗口左边的显示区域内显示出原始图像和用小波消噪后的图像。对一个图像如果用小波进行多层分解，则可对分解后的每一个高频系数进行阈值量化。每个高频系数层的阈值选择滚动条和文本框都按照层次大小，从上到下的顺序排列。在这里，可以选用一个名为Noissi2d的含噪声的图像(位于MATLAB\Toolbox\wavelet\wavedemo目录下)，将它用sym3小波进行2层小波分解，然后用手动软阈值和手动硬阈值分别进行消噪处理，其消噪结果分别如图2.125和图2.126所示。

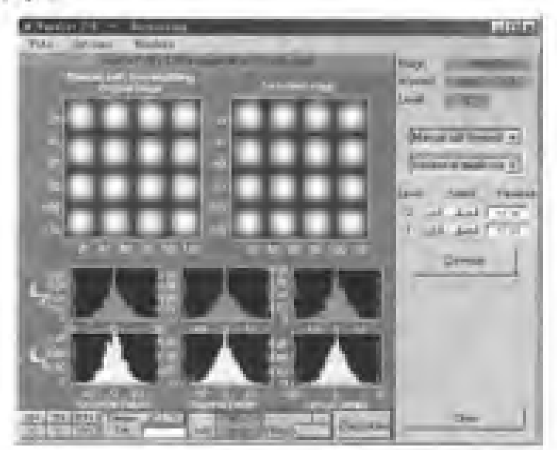




图2.125　软阈值消噪效果

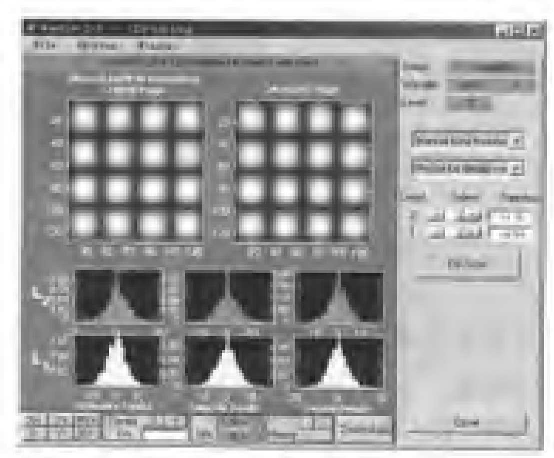


图2.126　硬阈值消噪效果

两种阈值量化后，就图像的消噪效果来看，用硬阈值量化进行消噪的图像还不是很平滑，而用软阈值量化进行消噪后的图像能将大部分噪声除去，消噪效果比用硬阈值的好。这是因为采用硬阈值，对高频系数进行量化时，产生了数值的突变，这种数值突变变成一种高频噪声信息，在进行图像重构时，又将这种噪声信息引入到图像中去了。

2.9.5　一维小波包图形工具简介

当用户单击小波工具箱主菜单窗口中的Wavelet Package 1－D按钮后，弹出的一维小波包图形工具窗口，在外观上与一维小波图形工具类似。左边是信号的显示区域，右边是对信号进行小波包分析的各种按钮和选择框。为了说明这一工具的使用，可以通过对一个名为Heavysin的信号进行实例分析。

(1) 调入这个一维信号后可以看到，在显示区域中有四个显示方框：Analyzed Singal、Colored Coefficients for Terminal Nodes、Decomposition Tree和Node Action Result。

・ Analyzed Singal显示方框在显示区域的右上角，这个方框显示的是原始信号的波形。

・ Colored Coefficients for Terminal Nodes显示方框在显示区域的右下角，这个方框显示的是信号小波包分解后系数的灰度图像。由于在一维小波包图形工具中，对一个信号是按照完整二叉树的方式来进行小波包分解的。若对一个信号进行n层小波包分解，则它的终结点个数为2^n个，该图形工具对第n层的2^n组分解系数以灰度图像的方式来显示其结果。在它的正下方，有一个Scale of Colors from Min To Max颜色条，它用来表示系数大小与颜色的对应关系。

· Decomposition Tree(小波包分解树结构)显示方框在显示区域的左上角。当信号经过小波包分解后，在这个方框中显示的是信号的小波包分解树结构。在它的上端和下端各有一个滚动条，上端滚动条的作用是放大或缩小小波包树结构图形，下端滚动条的作用是对小波包树结构图形进行左右移动。将上端的滚动条往右移，则放大图形，往左移则缩小图形；将下端的滚动条往右移，则图形往右移，反之则往左移。由于一个信号用小波包进行多层分解之后，树结构可能很复杂，所以，这两种滚动条可以使用户方便地查看各个结点的细节。

· Node Action Result 显示方框在显示区域的左下角，用户可以选择某个结点进行分析，然后将分析后的结果显示在该显示方框中。对结点具体的分析内容由 Node Action 选择框来决定。

(2) 在窗口的右边是对信号进行分析的各种选择框和按钮。选择框有小波类型和小波分解层次选择框，这两个选择框在每一个图形工具中都有，功能也都一样，即选择小波分解的类型和小波分解的层次。其它的选择框有：

· Entropy (熵标准)选择框，可以选择不同的熵标准，用于信号小波包最优树的选择、信号的压缩和消噪等，这些熵标准是 shannon、threshold、norm、log energy、sure 和 user，其选择如图 2.127 所示。各个熵标准的具体特性请参见本章第六节的有关内容。

· Cut Tree at Level 选择框，它可将小波包分解树剪切到某一层。如果对一个信号进行 n 层分解后，这一选择框可以选择从 0 到 n 之间的层次。若一个信号只需要分解到第 2 层就足以满足分析要求，那么就可以将它剪切到第 2 层。操作方法是选择数字 2 即可。

· Node Label(结点标识方式)选择框，结点的标识方式有 Depth - Pos(深度－位置)、Index(索引)、Entropy(熵值)、Opt. Ent(最优熵)、Length(长度)、None(无标识)和 Type(类型)这七种标识方式，具体如图 2.128 所示。

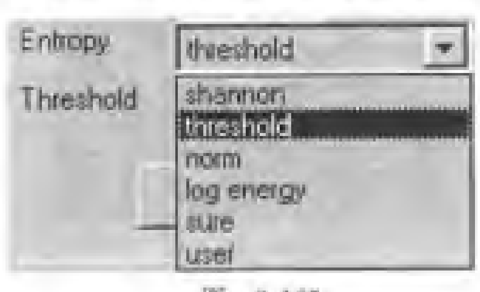



图 2.127

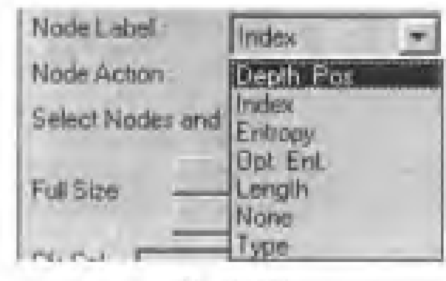


图 2.128

Depth - Pos 方式在结点处是以深度－位置形式的二维数组标注的；Index 方式是以结点的索引值来标识结点，结点的索引是从上到下、从左到右进行编号，根结点为零；Entropy 方式所标识的是该结点对应的熵值；Opt. Ent 方式所标注的是该结点是否是最优熵，如果不是最优熵，则标注为"NaN"(Not a Number)；Length 方式所标识的是该结点的数据长度；在 None 方式下，结点处没有任何标注；Type 方式所标识的是该结点所表示的是高频信息(d)还是低频信息(a)，对一个最简单的二叉子树，它的左边表示低频信息，用(a)表示，它的右边表示高频信息，用(d)表示。

· Node Action 选择框有 Visualize(显示)、Split/Merge(分解/合成)、Recons.(重构)、Select On(选中)、Select Off(取消选中)、Statistics(统计)和 View Col. Cfs(查看终结点系数值的颜色)这七种方式，具体如图 2.129 所示。

在 Visualize 方式下，如果单击树结构中的任意一个结点，则在 Node Action 显示方框中就会显示这一结点的系数曲线。在 Split/Merge 方式下，当单击树结构中的任意一个结点时，如果该结点是终结点，它对这个结点就进行下一层的小波包单层二叉分解，即 Split；如果该结点不是终结点，则它会将这一结点以下的所有子结点剪切，即 Merge，使这个结点变成终结点。在 Recons. 方式下，它将所选结点的系数进行重构，使重构后的长度与原始信号的长度一致。在 Select On 方式下，用鼠标单击树结构中的某一个或某几个结点进行结点选择(结点被选中时，颜色呈绿色)。当选中几个结点后，单击该选择框正下方的 Reconstruct 按钮，就可以对几个结点的分解系数进行重构。在重构操作完成之后，可以再选择该选择框中的 Select off 方式来取消已经选中的结点。在 Statistics 方式下，当单击树结构中的任意一个结点时，会弹出一个统计结果窗口，这个弹出窗口的内容和一维离散小波图形分析工具中的 Statistics 窗口类似，即可以将所选结点的数学特征进行统计计算，并以文本方式显示其结果，同时，将该结点的信息以直方图形式和求和形式的直方图显示出来。在 View Col. Cfs 方式下，只能查看小波包分解树结构的终结点的系数。此时，单击树结构中的任意一个终结点，则在 Colored Coefficients for Terminal Nodes 显示方框中显示出所选结点系数值对应的灰度图像，而其它结点的系数值灰度图像均不显示。

· Cfs. Col (系数染色)选择框。在这一选择框中，对系数的分类有 FRQ 和 NAT 两大类，其中 FRQ 表示系数值是以频率分布来度量的，NAT 表示系数值是以奈特(信息量的自然单位)来度量的。在每种度量单位下，都有 Global＋abs (全局＋绝对值)、By Level＋abs(层次＋绝对值)、Global(全局)和 by Level(按照层次)四种系数值的染色方式，具体如图 2.130 所示。

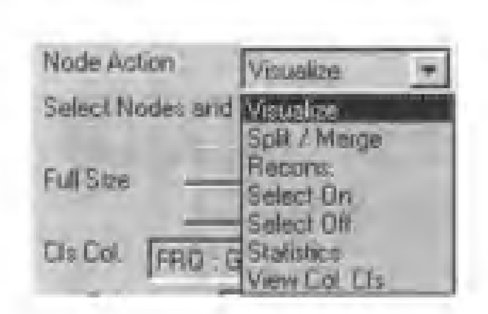


图　2.129

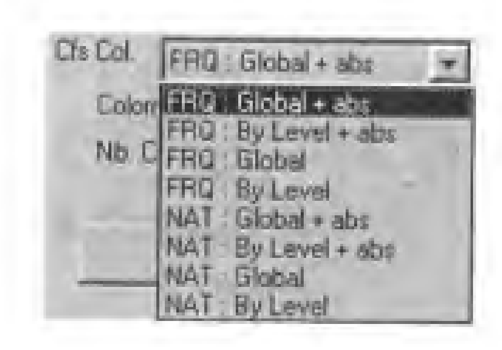


图　2.130

Global＋abs 染色方式是将原始信号及其所有小波包分解层的系数值、绝对值的范围作为颜色的染色范围。总的颜色数由 Nb. Colors 中的滚动条或它右边的文本框来决定。在这种方式下，在终结点的 Colored Coefficients for Terminal Nodes 显示方框中，可以很清楚地看到所有终结点和原始信号的数值分布情况。在 By Level＋abs 方式下，每个终结点的系数和原始信号绝对值的染色都是独立的，即每一层都有相同的 Nb. Colors(颜色数)。在这种情况下，可以很方便地看到各层的系数值分布情况。Global 方式和 by Level 方式与以上这两种方式的系数染色类似，唯一不同的是它所染色的依据不是绝对值，而是它本身值的大小。在以 NAT 为度量基准时，也有四种系数染色方式，这与以 FAQ 为度量基准的系数染色方式是一样的，只是染色基准有所不同。

· Colormap(颜色映射)选择框，可以将数值用不同的颜色映射模式来表示。当选择不

同的颜色映射模式时，系数值的显示效果是不同的。选择适当的颜色映射模式可以获得最好的显示效果。

(3) 在窗口的右上方，有七个分析按钮，分别是 Analyze（分析）、Compress（压缩）、De-noise（消噪）、Initial Tree（初始树）、Wavelet Tree（小波树）、Best Tree（最佳树）和 Best Level（最佳小波包分解层），具体显示如图 2.131。

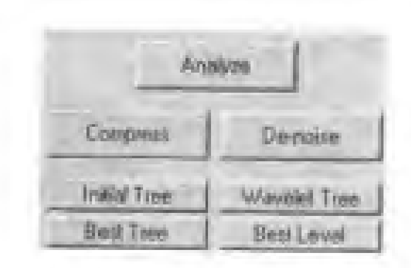


图　2.131

• Analyze 按钮执行的操作是对信号进行小波包分解，然后将分解结果显示在窗口中。

• Initial Tree 按钮执行的操作是计算一个信号的小波包分解的初始树并显示。

• Wavelet Tree 按钮执行的操作是计算一个信号的小波分解并显示。

• Best Tree 按钮执行的操作是在计算一个信号的小波包分解之后，用熵标准计算出最佳的小波包分解树。熵标准的选择由 Entropy 选择框来实现。

• Best Level 按钮执行的操作是根据熵标准计算出最佳的小波包分解层次，并显示。

• Compress 按钮执行的操作是对所分析的信号在小波包分解之后，对各层的高频系数进行阈值量化处理，再重构信号，从而对信号用小波包方法进行压缩。进行阈值量化时，熵标准由 Entropy 选择框来选择。在一维小波包图形工具中，只有全局阈值，当阈值和熵标准都选定之后，单击 Compress 按钮，就可以对该信号进行压缩。信号压缩后，压缩信号（黄色）与原始信号（红色）在同一显示方框中显示，并且阈值量化后的系数与量化前的系数也在显示区域中对比显示。图 2.132 是信号压缩后的效果显示。

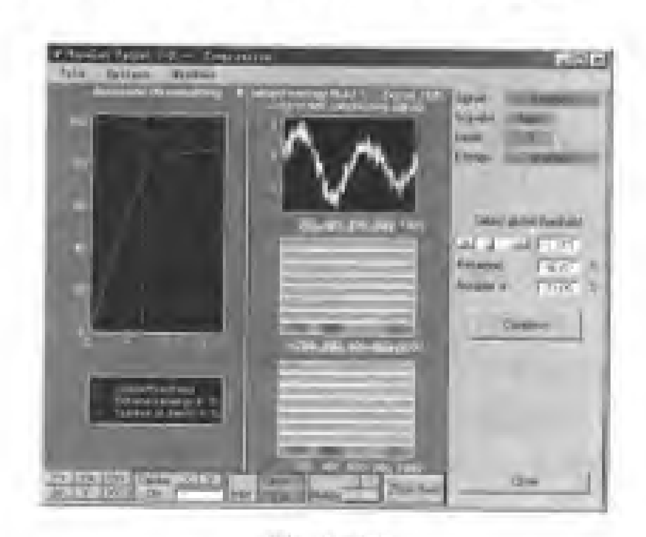


图　2.132

• De-noise 按钮执行的操作是去除信号中的高频干扰，用小波包方法实现消噪。进行阈值量化时，熵标准也由 Entropy 选择框来选择。单击此按钮后，弹出一个小波包信号消噪窗口，在窗口的右边有 Automatic Hard Thresholding 和 Automatic soft Thresholding 选择框，可以选取硬阈值方式或软阈值方式进行信号的消噪。在窗口的右下方，有一个 Number of bins（柱条数）文本框，可以输入数值来确定信号高频系数直方图中的柱条数，信号高频系数的直方图有助于我们更直观地观察信号中噪声的分布情况。在阈值和熵标准都选定之

后，单击Compress按钮，就可以对信号进行消噪。信号消噪后，消噪信号(黄色)与原始信号(红色)在同一显示方框中显示，并且阈值量化后的系数与量化前的系数也在显示区域中对比显示。图2.133是信号消噪后的效果显示。

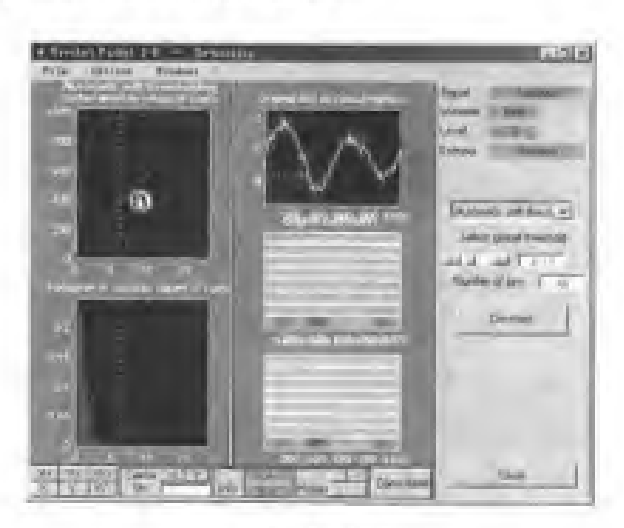


图　2.133

2.9.6　二维小波包图形工具简介

当单击小波工具箱主菜单窗口中的Wavelet Package 2—D按钮后，弹出Wavelet Package 2—D(二维小波包图形工具)窗口，它在外观上与一维小波包图形工具类似。左边是图像的显示区域，右边是对图像进行小波包分析的各种按钮和选择框。

(1) 在调入这个图像后，在显示区域中有四个显示方框，分别是Analyzed Image(待分析图像)、Colored Coefficients for Terminal Nodes(终结点的系数染色)、Decomposition Tree(小波包分解树结构)和Node Action Result(结点处理后的结果)。这四个显示区域所显示的内容与Wavelet Package 1—D窗口中显示的内容很相似。

Analyzed Image方框中显示的是原始图像。Decomposition Tree方框中显示的是图像小波包分解的四叉树分解图像。Node Action Result方框中显示的是所选结点的分析结果，对结点具体的分析内容由Node Action(结点处理)选择框来决定。

(2) 在窗口的右边是对图像进行分析的各种选择框按钮。

在这些选择框中，有两个通用的选择框，即Wavelet选择框和Level选择框。这两个选择框在每一个图形工具中都有，功能也都一样，可以通过它们选择小波函数和小波包分解的层数。

另外，还有几个选择框，它们分别是Entropy选择框、Cut Tree at Level选择框、Node Label选择框、Node Action选择框、Cfs. Col选择框、Colormap选择框。

(3) 在窗口的右上方，有七个分析按钮，分别是Analyze(分析)、Compress(压缩)、De-noise(消噪)、Initial Tree(初始树)、Wavelet Tree(小波树)、Best Tree(最佳树)和Best Level(最佳小波包分解层)。

二维小波包图形工具的用法与一维小波包图形工具的用法基本相同，在此不作详述。

2.9.7 小波和小波包信息显示工具简介

在小波工具箱主菜单窗口中，有 Wavelet Display(小波信息显示)和 Wavelet Packet Display(小波包信息显示)两个按钮，它们分别显示小波函数和小波包函数簇的信息。

(1) 小波信息显示工具

在一般情况下，每一个小波函数都有对应的尺度函数。在不同的尺度下，信号在小波函数上的投影系数就是信号不同频带的小波分解系数。小波函数与相应的尺度函数是一个线性关系，它都是由最初的一个尺度函数经过线性迭代运算得来的。设 $\phi(x)$ 是一个最初的尺度函数，那么小波函数与尺度函数就有如下的关系(双尺度方程)：

$$\phi(x) = \sqrt{2}\sum_{k=0}^{2N-1} h(k)\phi(2x-k)$$

$$\Psi(x) = \sqrt{2}\sum_{k=0}^{2N-1} g(k)\phi(2x-k)$$

式中 $h(k)$ 与 $g(k)$ 都是长度为 $2N-1$ 的滤波器，它们在数学上都是系数序列，在物理意义上，$h(k)$ 是一个低通滤波器，$g(k)$ 是一个高通滤波器。

单击 Wavelet Display 按钮后，弹出一个 Wavelet Display 窗口，它的右边有 Wavelet 选择框和 Refinement 选择框。Wavelet 选择框可以选择所要显示的小波函数；Refinement 选择框用于控制计算小波函数的运算精度，数值越大，表示运算迭代次数越多，小波函数的精度越高。在窗口右边的 Information on 标签下有两个按钮，它们分别是 X Wavelet 按钮和 Wavelets 按钮。X Wavelet 按钮以文本方式显示所选择小波函数的各项特征，如缩写名、是否具有正交性(双正交性)、是否具有紧支撑、有效支撑长度、是否具有对称性等。"X"表示所选择的小波函数的函数名，它随着小波函数的变化而变化。Wavelets 按钮以文本方式简要显示在 MATLAB 工具箱中所有小波函数的对比特性。

在这里，我们选择 db3 小波函数，图 2.134 显示的是小波函数的波形、尺度函数的波形式以及相应的滤波器(包括低通和高通、分解和重构滤波器)。

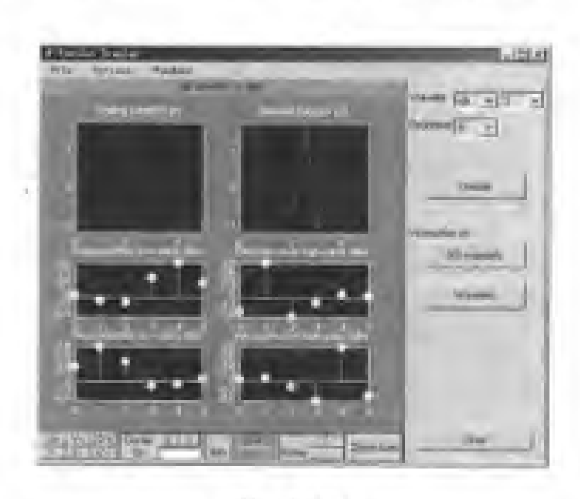


图 2.134

(2) 小波包信息显示工具

小波包函数簇是以小波函数为基础产生的，函数簇中的函数实际上就是各个带通滤

波器。

单击 Wavelet Packet Display 按钮后，弹出一个 Wavelet Packet Display 窗口，它的右边有 Wavelet 选择框、Refinement 选择框和 Wav. Pack from 0 to：选择框。Wavelet 选择框可以选择所要显示的小波函数；Refinement 选择框用于控制计算小波函数的运算精度，数值越大，表示运算迭代次数越多，小波函数的精度越高；Wav. Pack from 0 to：选择框可以选择所要显示的小波包基函数的个数。在窗口右边的 Information on 标签下，也有两个按钮，它们分别是 X Wavelets 按钮和 W Systems 按钮。X Wavelets 按钮以文本方式显示所选择小波包函数的各项特征，如缩写名、是否具有正交性(双正交性)、是否具有紧支撑、有效支撑长度、是否具有对称性等。"X"表示所选择的小波包函数的函数名，它随着小波包函数的变化而变化。W Systems 按钮以文本方式简要显示在 MATLAB 工具箱中所有小波包函数的对比特性。

在这里，我们选择 db3 小波包函数，显示由它生成的七个小波包基函数，其结果如图 2.135 所示。它显示的是各个小波包基函数的波形。

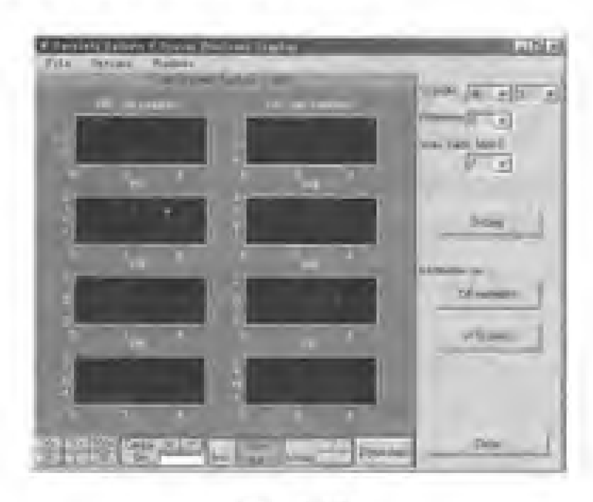


图　2.135

2.10　小波分析中的数据 I/O 函数

在 MATLAB 中，用户对各种数据进行小波分析时，需要从外界读取大量的一维或二维数据，同时在分析之后又需要将这些数据存储起来。但是，由于数据的格式具有多样性，不同的数据格式在应用之中存在着一定的局限性。所以，在 MATLAB 中数据一般是以 *. mat 格式存储的，它只能适用于 MATLAB 工作环境，而其它语言(如 C 语言)无法对这种数据进行操作，因而这种格式的数据不具有跨平台性。为此，MATLAB 工具箱提供了多种数据输入输出函数(即 I/O 函数)，用户既可以按照 MATLAB 的 *. mat 文件格式存储或读取数据，也可以按照用户自己指定的数据格式存储或读取，并将这些数据用于其它不同的应用程序中去，例如 C(或 C++)语言或 Fortran 语言等，因而大大拓宽了 MATLAB 的应用范围。在这一节里，我们分别对一维小波分析和二维小波分析中的数据 I/O 进行介绍。

在 MATLAB 中，I/O 函数主要有：load 、save、fopen、fclose、fscanf、fread、fprinif、fwrite、ftell、fseek、ferror、imread 和 imwrite。

1. load

功能：读取数据

格式：① load

② load filename

③ load (filename)

④ load filename. ext

⑤ load filename -ascii

⑥ load filename -mat

说明：load 函数是进行数据读取。格式①读取名为'matlab. mat'的数据文件(参见 save)；格式②读取一个 *. mat 格式的数据，其中，'filename. mat'为数据文件名；格式③读取一个文件名为 filename 的数据，例如：

str='filename. mat'；

load (str)；

上述这三种格式读取的是一个二进制文件。

格式④读取一个 ASCII 格式数据。数据之间用空格分开。在 ASCII 文件中，可以包含 MATLAB 的注释行语句，即以"%"开头的语句。格式⑤或格式⑥可以强制将数据文件以 ASCII 文件或 MAT 文件形式处理。另外，在使用 load 函数时，我们可以给出数据的路径全名。

Load 函数既可以读取 *. mat 格式文件，也可以读取文本文件数据，文本文件数据必须按照一个矩形格式来组织，即各个数据之间用空格分开，每行数据的个数要相等。例如：在 MATLAB 之外，创建了一个文本文件数据，它包含以下四行：

```
16.0    3.0    2.0   13.0
 5.0   10.0   11.0    8.0
 9.0    6.0    7.0   12.0
 4.0   15.0   14.0    1.0
```

如果这些数据已经存在一个 magik. dat 文件中，则输入以下的命令行进行数据读取：

load magik. dat

举例：

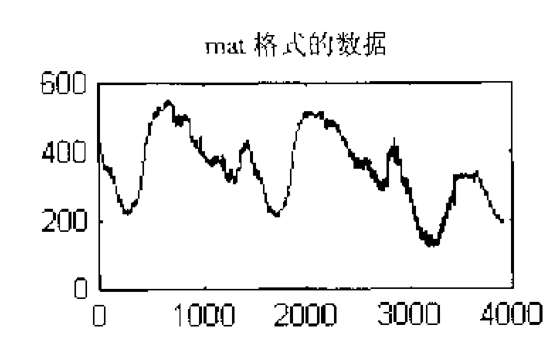
例 1：

读取一个 *. mat 格式的一维数据文件和一个文本格式的数据文件，并显示其内容。

```
%读取一个*.mat格式的一维数据
load leleccum;
s1=leleccum(1:3920);
%显示*.mat文件数据
subplot(3,2,1); plot(s1);
title('mat格式的数据');
%读取一个文本格式的一维数据
load d:\zjb\204data\li1\vg20.dat;
```

```
s2=vg20;
%显示文本文件数据
subplot(3,2,2); plot(s2);
title('文本格式的数据')
```

输出结果(如图 2.136 所示):



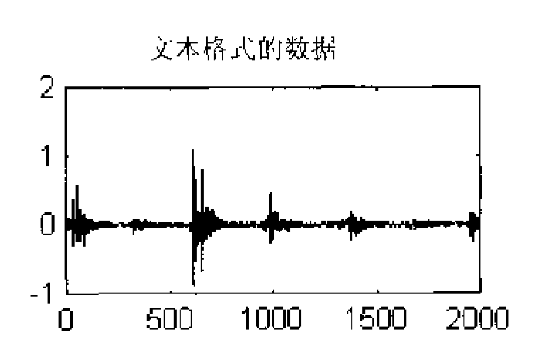


图 2.136

例 2: 装入一个 mat 格式的二维图像数据，它位于 Toolbox\wavelet\wavedemo 目录下，名为 wbarb. mat。

load wbarb

参见: save

用户可以创建自己的包含矩阵的 M 文件，这个文件可以由文本文件来创建。创建方法很简单，只要在 MATLAB 的命令行中用 save 命令函数输入相同的文件名，并以. m 结尾即可。例如，对上面建立的文本文件 magik. dat，在 MATLAB 中输入以下的命令行：

save magik. m

则将该文件存成了 M 文件。

2. save

功能: 将 MATLAB 工作环境中的数据存盘

格式: ① save

② save filename

③ save filename variables

④ save filename options

⑤ save filename variables options

说明: 格式①是将当前工作环境下的所有数据存入一个名为 matlab. mat 的二进制类型的文件。当用 clear 命令清除当前 MATLAB 环境中的所有变量时，可以用 load 函数读取该 matlab. mat 文件，即 load matlab；格式②在功能上与格式①类似，它将当前工作环境下的所有数据以用户指定的文件名 filename. mat 存盘，通过 load 命令可以调用该文件；格式④是在存盘时，按照 options 中指定的数据格式存储。options 可以进行不同的选择，具体见表 2－7。

表 2－7　Options 参数选择

Options	数据存储形式
－ascii	8 位 ASCII 数据格式
－ascii －double	16 位 ASCII 数据格式
－ascii －tabs	8 位 ASCII 数据格式，数据表单分离(tab－seperated)
－ascii －double －tabs	16 位 ASCII 数据格式，数据表单分离(tab－seperated)

数据保存到文件中的排列格式与数据在变量中的排列格式一致，如变量是列向量时，数据写到文件后以列的形式排列。通常来说，我们应把一类数据写成一列。

在数据存成 ASCII 格式后，当用 load 读取这个文件时，数据由 ASCII 格式变成实数数据格式，这时，用户还可以用冒号来读取其中的某些数据。但是，如果用－ascii 格式来存储复数，在读取数据时，由于 MATLAB 不能读取虚数的符号 i，所以复数的虚数部分将会丢失。

通过 save 函数进行数据存储时，数据的格式由待存储的数据数组的大小和类型来决定。当数组中存在小数，并且数组的长度小于或等于 10 000 时，则它以浮点数的格式存储，每个数据的长度是 8 个字节(byte)。如果数组中的数据都是整数，并且数组的长度大于 10 000，则它以表 2－8 中的格式存储，这时数据可以占用更少磁盘空间。

表 2－8　数 据 的 范 围

数据的范围	每个数据元素的长度(字节数)
0～255	1
0～65535	2
－32267～32267	2
$2^{31}+1$～$2^{31}-1$	4
其它	8

举例：

例 1　对一个一维信号进行小波分解后，将分解系数存盘，并显示存盘后的数据，验证其是否正确。

```
%读取一个一维信号数据
load leleccum; s=leleccum(1:3920); ls=length(s);
[ca1,cd1]=dwt(s,'db1');
%将用 db1 小波分解后的系数存盘
%存储低频分解系数，文件名为 ca1.mat
save ca1;
%存储高频分解系数，文件名为 cd1.mat
save cd1;
```

```
%重构第一层的低频部分 a1
a1=upcoef('a',ca1,'db1',1,ls);
%重构第一层的高频部分 d1
d1=upcoef('d',cd1,'db1',1,ls);
%存储低频和高频部分，这时在 c:\temp 目录中有 a1.mat 和 d1.mat 文件
save c:\temp\a1;
save c:\temp\d1;
%从磁盘文件中装入所存储的数据，并显示
load ca1; s1—ca1;
load cd1; s2=cd1;
load c:\temp\a1; s3=a1;
load c:\temp\d1; s4=d1;
subplot(221); plot(s1); title('ca1.mat 文件中的数据');
subplot(222); plot(s2); title('cd1.mat 文件中的数据');
subplot(223); plot(s3); title('a1.mat 文件中的数据');
subplot(224); plot(s4); title('d1.mat 文件中的数据')
```

输出结果(如图 2.137 所示)：

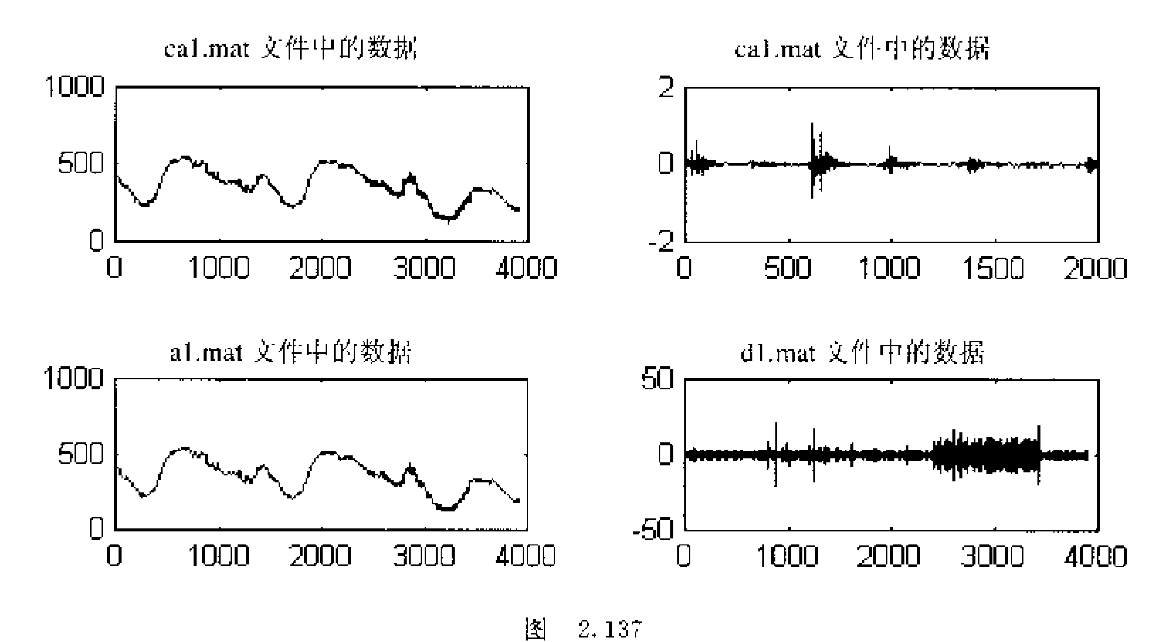


图 2.137

例 2：对一个二维图像进行小波分解后，将分解系数存盘，然后从文件中读取各个数据，并显示。

```
clear;
load wbarb;
%用 bior3.7 小波进行单尺度分解
[ca1,ch1,cv1,cd1]=dwt2(X,'bior3.7');
%存储二维小波分解图像后的系数
save myimage;
```

```
a1=upcoef2('a',ca1,'bior3.7',1);
h1=upcoef2('h',ch1,'bior3.7',1);
v1=upcoef2('v',cv1,'bior3.7',1);
d1=upcoef2('d',cd1,'bior3.7',1);
%在 MATLAB 中，图像分解完成后，可以直接存储成 *.mat 文件，
%也可以存成 wa2 文件,其后缀名是 *.wa2(Wavelet Analysis 2-D)
save myimage.wa2;
%清除 MATLAB 当前工作环境中的所有变量
clear;
disp('显示清除后 MATLAB 工作环境中的变量');
whos
%读取上面所存储的 myimage.wa2 数据
load myimage.wa2 -mat;
disp('显示读取数据后 MATLAB 工作环境中的变量');
whos
```

输出结果：

显示清除后 MATLAB 工作环境中的变量

显示读取数据后 MATLAB 工作环境中的变量

Name	Size	Bytes	Class
X	256x256	524288	double array
a1	286x286	654368	double array
ca1	135x135	145800	double array
cd1	135x135	145800	double array
ch1	135x135	145800	double array
cv1	135x135	145800	double array
d1	286x286	654368	double array
h1	286x286	654368	double array
map	192x3	4608	double array
v1	286x286	654368	double array

Grand total is 466196 elements using 3729568 bytes

参见：load

3. fopen

功能：打开一个文件或获取一个文件的信息

格式：① fid=fopen(filename,permission)

② [fid,message]=fopen(filename,perimission,format)

③ fids=fopen('all')

④ [filename,perimission,format]=fopen(fid)

说明：参数 fid 是一个文件标识参数，如果 fopen 成功地打开了一个文件，则返回一个正整

数；如果 fopen 打开一个文件失败，则 fid 返回 −1。这时，消息参数 message 返回出错信息。

格式①以 permission 模式打开一个名为 filename 的文件，并将文件标识返回给 fid。名为 filename 的文件，是一个在 MATLAB 环境中的包含路径名的文件，如果以读取方式打开这个文件，而这个文件在当前工作路径中没有找到，则 fopen 沿着 MATLAB 的搜索路径进行搜寻。

permission 参数可进行如下选择：

- 'r'　用只读方式打开一个文件(默认的方式)；
- 'r+'　用读写方式打开一个文件；
- 'w'　删除已有文件的全部内容或创建一个新文件(如果该文件名不存在)，打开后写入数据；
- 'w+'　删除已有文件的全部内容或创建一个新文件(如果该文件名不存在)，打开后写入或读取数据；
- 'a'　创建并打开一个新文件，或打开一个已有的文件，在文件的末端添加内容；
- 'a+'　创建并打开一个新文件，或打开一个已有的文件，在文件的末端添加内容或读取内容；

在这些字符串后面添加一个't'，例如，变成'rt'，则将该文件以文本格式打开。在 DOS 和 VMS 操作系统中，用户只有将文件的 permission 参数设置成'rt'之后，才能读取文本格式的数据。与此类似，在这些字符串后面添加一个'b'，则系统就会将文件以二进制格式打开，这是一种默认情况。

格式②与格式①所执行的操作一样，但它除返回文件标识之外，还返回文件标识的消息变量 message。另外，在格式②的输入参数中，format 参数用于定义数据格式，它可以使用户的文件在不同的计算机之间共享。如果将 format 参数忽略，则使用的格式是当前计算机的数据格式。用户可以在 fread 和 fwrite 函数中重载格式②中已经定义的数据格式。这些数据格式有：

'native'或'n'　当前计算机的数据格式

'ieee−le'或'b'　IEEE 浮点数

格式③是一个行向量，它包括所有已打开文件的文件标识。该向量的元素个数与已打开的文件的个数相等。

格式④返回的是 filename 的全部字符串(包括文件名和路径名)、permission 字符串和 format 字符串。如果输入参数 fid 是一个无效的参数，则返回的三个参数均为空字符串。在格式④中，permission 和 format 可以省略。

举例：

```
%读取一个已有的数据
disp('读取一个已有的数据');
[fidd,msg]=fopen('d:\zjb\204data\li1\vg20.dat')
[filename,perimission,format]=fopen(fidd)
disp('读取一个不存在的数据');
[fidd,msg]=fopen('d:\zjb\204data\li1\vg201.dat')
```

输出结果：

```
读取一个已有的数据
fidd =
     3
msg =
     ''
filename =
d:\zjb\204data\li1\vg20.dat
perimission =
rb
format =
ieee-le

读取一个不存在的数据
fidd =
     -1
msg =
Cannot open file. Existence? Permissions? Memory? . . .
```

参见：fclose，fscanf，fread，fprintf，fwrite，ftell，fseek，ferror

4. fclose

功能：关闭一个或多个已打开的文件

格式：① status=fclose(fid)

　　② ststus=fclose('all')

说明：格式①关闭一个指定的文件。如果执行成功，返回 0；如果执行不成功，则返回－1。输入参数 fid 是由 fopen 函数所返回的文件标识。格式②用于关闭所有已经打开的文件，如果执行成功则返回 0，否则返回－1。

举例：

```
%读取一个已有的数据
disp('读取一个已有的数据');
[fidd,msg]=fopen('d:\zjb\204data\li1\vg20.dat')
%关闭该文件
sta=fclose(fidd)
```

输出结果：

```
读取一个已有的数据
fidd =
     3
msg =
     ''
sta =
     0
```

参见：fopen，fscanf，fread，fprintf，fwrite，ftell，fseek，ferror

5. fprintf

功能：将数据以一定的格式写到文件中

格式：① count=fprintf(fid,format,A)

② fprintf(format,A)

说明：对于格式①，它将变量A中的数据按照参数format所定义的格式写入到文件标识为fid(参见fopen)的文件中去，如果A中的数据为复数时，则它只将实数部分写入到文件中去。参数fid是由函数fopen返回的一个文件标识，它是一个整数。当fid=1时，它就向屏幕输出数据；当fid=2时，它进行标准错误处理。参数count返回所写入文件的数据个数。当参数fid没有显式定义时(格式②的情况)，它以默认的形式将变量A中的内容输出到屏幕上，即fid=1的情况。

参数format用于定义数据的格式，它是一个字符串，由起始标志符、数据左右排列方式、符号位、数据宽度和其它的输出格式字符组成。它可以包含普通的字母、数字、(alphanumeric charactics)、escape字符、数据转化字符等，例如：

%－12.5e

这个函数与ANSI C语言中的fprintf()函数很相似，但有以下几点不同：

(1) MATLAB支持%o、%u、%x和%X这几种非标准的二级格式指定。另外，对于t，基本的C数据类型是浮点数，而不是无符号整数；对于d，基本的C数据类型是双精度浮点数，而不是无符号整数。例如，要想将一个双精度数以十六进制数的格式输出，格式形式为：'%bx'。

(2) 函数fprintf是按照列的顺序来读取矩阵A中的数据，每次读取矩阵A的一个数据时，都转换成format所定义的数据格式，直到数据全部读完。

表2－9中列出了在参数format中出现的非字母数字字符(non-alphanumeric characters)的形式及含义。

表2－9　Escape字符

字　符	说　　明
\n	换行
\t	水平方向的Tab
\b	退后一格
\r	回车
\f	走纸换页
\\	反斜线
\"or"	单引号
%%	百分号

通过数据转换可以定义数据输出格式，数据转换格式详见表2－10。

表 2-10　数据转换格式

字　符	说　　明
%c	单个字符
%d	十进制数据(有符号)
%e	指数形式的浮点数(用小写的 e，如 3.141e+00)
%E	指数形式的浮点数(用大写的 e，如 3.141E+00)
%f	固定小数点的浮点数
%g	%e 或%f 的紧促形式，它将数据中无意义的零省略了
%G	与%g 的功能一样，用的是大写的 G
%o	八进制数据(无符号)
%s	字符串
%u	十进制浮点数(无符号)
%x	十六进制数(字母 a～f 用小写)
%X	十六进制数(字母 A～F 用大写)

可以在数据转换字符与“%”之间加入附加字符，其具体情况见表 2-11。

表 2-11　附 加 字 符

字　符	说　　明	举例
负号(—)	在输出数据中左对齐	%-5.2d
正号(+)	在输出数据中始终打印符号(+或-)	%+5.2d
零(0)	在输出数据中宽度不够的补零，而不是用空格	%05.2d
数字(数据宽度)	一个数字字符串，指定在输出数据中所打印的最少的数字字符数	%6f
数字(精度)	一个数字字符串加上一个点号(.)，指定在输出数据中所打印的数字字符数和点号右边的数字字符数	%6.2f

举例：

例 1：产生一组浮点数，并用 fprintf 进行格式存储

```
%产生一组浮点数
x=0:0.1:1;
y=[x;exp(x)];
%打开一个文件
fid=fopen('c:\temp\exp.txt','w');
%把数据写入该打开的文件中
fprintf(fid,'%6.2f %12.8f\n',y);
```

fclose(fid); %关闭文件

当执行完以上的程序之后，可以看到在 c:\temp 目录下，生成了一个名为 exp.txt 的文本文件，它的内容为：

```
0.00    1.00000000
0.10    1.10517092
0.20    1.22140276
0.30    1.34985881
0.40    1.49182470
0.50    1.64872127
0.60    1.82211880
0.70    2.01375271
0.80    2.22554093
0.90    2.45960311
1.00    2.71828183
```

例 2：将一个字符串以标准输出方式输出，即输出到屏幕上。

fprintf('一个单位圆的周长是 %g。\n',2 * pi)

则在屏幕上显示的是：

一个单位圆的周长是 6.283186。

fprintf(1,'今天是星期五。\n')

则在屏幕上显示的是：

今天是星期五。

此时，fprintf 函数的功能与 disp 函数一样。

例 3：将矩阵和向量混合在一起进行数据输出。

B=[8.8 7.7; 8800 7700];

fprintf(1,'X 是 %6.2f 米或 %8.3f 毫米。\n',9.9,9900,B)

则屏幕上的输出是：

X 是 9.90 米或 9900.000 毫米。

X 是 8.80 米或 8800.000 毫米。

X 是 7.70 米或 7700.000 毫米。

将矩阵和向量混合在一起进行数据输出时，向量是按照从左到右的顺序输出，矩阵是按列的顺序输出。

例 4：在 MATLAB 中，将双精度数据转换成整数。在这里，是将一个有符号的 32 比特数据转换成十六进制数。

a=[6 10 14 44];

fprintf(1,'%9X\n',a+(a<0) * 2^32)

则屏幕上的输出为：

```
6
A
E
```

2C

参见：fopen，fclose，fscanf，fread，fwrite，ftell，fseek，ferror

6. fwrite

功能：将数据按照二进制格式写到文件中

格式：① count=fwrite(fid,A,precision)

② count=fwrite(fid,A,precision,skip)

说明：对于格式①，它按照参数 precision 所定义的数据格式，将矩阵 A 中的数据写入到一个指定的文件中去，数据按照列的顺序写入。参数 count 返回成功写入到文件中的数据个数。参数 fid 是函数 fopen 返回的一个文件标识。

对于格式②，参数 skip 表示的是每写入一个数据之后，需要跳过的字节数(byte)。当一个数据记录需要不连续地插入数据时，利用该参数即可实现。如果参数 precision 是以比特形式给出的(例如是'bitN'或'ubitN')，则参数 skip 也以比特形式给出。

参数 precision 用于定义数据格式，其用法同函数 fread 中的 precision 参数一样。详见表 2-9、表 2-10 和表 2-11。

举例：

例 1：用 fwrite 函数创建一个二进制文件，它包括 25 个数据，以 5×5 的方阵存放，每个数据是长度为 4 个字节的整数。

```
fid=fopen('c:\temp\magic5.bin','wb');
fwrite(fid,magic(5),'integer * 4')
```

例 2：将一个文本格式浮点数的数据文件读取进来，然后再转化成二进制格式存盘。

```
%打开一个已存在的文本数据文件
fr=fopen('c:\temp\txtdata.txt','rt');
%从文件中读取数据
[a,count]=fread(fr,inf,'float32'); %inf 表示读到文件末尾
fclose(fr); %关闭文件
%打开一个待写入数据的文件
fw=fopen('c:\temp\bindata.txt','wb');
%将数据写入到文件中
fwrite(fw,a,'real * 4');
fclose(fw); %关闭文件
```

此程序执行完毕之后，在 c:\temp 目录下产生了一个二进制的数据文件。

参见：fopen，fclose，fscanf，fread，fprintf，ftell，fseek，ferror

7. fscanf

功能：从文件中按指定的格式读取数据

格式：① A=fscanf(fid,format)

② [A,count]=fscanf(fid,format,size)

说明：格式①读取由 fid 标识的文件中的所有数据。它将所读取的数据按照 format 中的格

式转化之后，再返回给矩阵A。输入参数fid是由fopen返回的一个文件标识；format是一个字符串，用于指定所读取数据的格式。

格式②只读取部分数据，数据的个数由size参数决定。count参数返回成功读取的数据的个数。

size参数可有以下几种选项：

- n　将n个数据读入到一个列向量中；
- inf　读到文件的末尾，返回的列向量中的元素个数与文件中的元素相等；
- [m,n]　读取数据并存入m×n矩阵(以列向量的顺序存入矩阵A)，n可以是inf，但m不能用inf。

在MATLAB的fscanf命令函数和C语言中，scanf()和fscanf()函数有一个很大的不同，即在MATLAB的fscanf是将读取的数据按照列的顺序存入矩阵的。

当MATLAB读取一个指定的文件时，它将文件中的数据与format参数中指定的格式相匹配。如果匹配成功，则将这个数据以列的顺序存入矩阵；如果只有前面部分数据匹配成功，则只将匹配成功的数据写入矩阵，这时，读取数据停止。format的字符串由普通的字符和数据转化格式符组成。例如：

%−12.5e

在这里，format的数据格式与C语言中的很类似，"%"是一个起始字符，"−"是正负数的标识，"12.5"是数据的字宽和精度，"e"是数据转换类型字符(在这里表示转换成指数形式)。

数据转换字符有以下几种形式：

%c	字符形式，由数据字宽来指定字符的个数
%d	十进制数据
%e,%f,%g	浮点数
%i	有符号整数
%o	有符号八进制整数
%s	一组连续字符串(中间无空格)
%u	有符号十进制整数
%x	有符号十六进制整数

当用%s时，一个字符串在MATLAB中用一个矩阵的几个元素存放，一个元素存放一个字符。用%c可以读取空格，而用%s可将字符串中的所有空格去除。另外，在以上这些数据转换字符中，可以加入字符'h'(short)或'l'(long)，例如，%hd表示短整数，%ld表示长整数。

举例：假如存在一个ASCII的文本文件，名为exp.txt，它的内容为

```
0.00    1.00000000
0.10    1.10517092
...
1.00    2.71828183
```

现在要将这一个文件读入到一个两列的MATLAB矩阵中去，读取程序为

%程序清单

```
fid=fopen('exp.txt');
a=fscanf(fid,'%g %g',[2 inf]);        %矩阵 a 有两行
a=a';                                 %矩阵的转置运算
fclose(fid)
```

运行以上程序后，矩阵 a 就是一个 10×2 的矩阵。

参见：fopen，fclose，fread，fprintf，fwrite，ftell，fseek，ferror

8. fread

功能：从二进制文件中读取数据

格式：① [A,count]=fread(fid,size,precision)

② [A,count]=fread(fid,size,precision,skip)

说明：对于格式①，它是从指定的二进制数据文件中读取二进制数据，并将数据写入变量 A 中。参数 count 返回成功读取数据的个数。参数 fid 是一个由函数 fopen 返回的文件标识。参数 size 定义读取数据个数，如果此参数没有给出显式定义，则用默认形式，即读取文件的全部数据(一直读到文件的末尾)。参数 size 有以下几种选项：

- n　从二进制文件中读取 n 个数据；
- inf　读取文件的全部数据；
- [m,n]　读取 m×n 个数据，并把这些数据存放在一个 m×n 矩阵中(以列的顺序存入数据)，n 可以是 inf，但 m 不能用 inf。

参数 precision 定义读取数据的数据格式。它也可以用"n * [数据格式]"的形式来定义某种格式的数据个数，其中，n 为正整数，此时 size 参数可忽略。例如，'40 * uchar'表示是 40 个无符号整数(相当于 size=40、precision 为'uchar')。如果 precision 没有给出显式定义，则它默认为'uchar'(无符号 8 比特字符)。

对于格式②，参数 skip 表示的是每读取一次数据之后，需要跳过的字节数(byte)。对于一个给定的数据记录，我们只需要其中的部分数据，并且这些数据不是连续排列的，则调用这个参数就变得十分有效。如果参数 precision 是以比特形式给出的，例如是'bitN'或'ubitN'，则参数 skip 也以比特形式给出。

对于不同类型的计算机，由于它们对同精度数据的表示方式有所不同，当用户需要将数据跨计算机(一般是指不同类型的计算机)使用时，表 2-12 给出了与计算机类型无关的数据格式。

表 2-12　MATLAB 中与计算机类型无关的数据格式

MATLAB	C 或 Fortran	说　明
'char'	'char * 1'	字符，长度为 1 字节
'shar'	'signed char'	有符号字符，长度为 1 字节
'uhar'	'unsigned char'	无符号字符，长度为 1 字节
'int8'	'integer * 1'	整数，长度为 1 字节

续表

MATLAB	C 或 Fortran	说　明
'int16'	'integer * 2'	整数，长度为2字节
'int32'	'integer * 4'	整数，长度为4字节
'int64'	'integer * 8'	整数，长度为8字节
'uint8'	'integer * 1'	无符号整数，长度为1字节
'uint16'	'integer * 2'	无符号整数，长度为2字节
'uint32'	'integer * 4'	无符号整数，长度为4字节
'uint64'	'integer * 8'	无符号整数，长度为8字节
'float32'	'real * 4'	浮点数，长度为4字节
'float64'	'real * 8'	浮点数，长度为8字节

下面，我们给出在MATALB中与计算机类型有关的数据格式，具体见表2-13：

表2-13　MATLAB中与计算机类型有关的数据格式

MATLAB	C 或 Fortran	说　明
'short'	'short'	短整数，长度为2字节
'int'	'int'	整数，长度为4字节
'long'	'long'	长整数，长度为4或8字节
'ushort'	'unsigned short'	无符号短整数，长度为2字节
'uint'	'unsigned int'	无符号整数，长度为4字节
'ulong'	'unsigned long'	无符号长整数，长度为4或8字节
'float'	'float'	浮点数，长度为4字节
'double'	'double'	双精度数，长度为8字节

另外，在MATLAB中，还可以用比特(bit)的形式来定义输入数据的格式，而不用字节(byte)的形式定义，具体见表2-14：

表2-14　MATLAB中以比特形式定义的数据格式

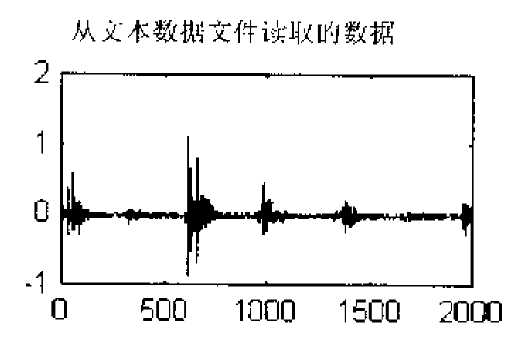
MATLAB	C 或 Fortran	说　明
'bitN'		无符号整数，长度为N比特(1≤N≤64)
'ubitN'		有符号整数，长度为N比特(1≤N≤64)

举例：从一个二进制的文件中读取数据(该文件中存放的是4个字节长度的浮点数)，并显示其结果。

```
%打开一个文本数据文件
fr=fopen('d:\zjb\204data\li1\vg20.dat','rt');
%把文本数据文件中的数据读到一个向量a中，count记录数据的个数
```

```
[a,count]=fscanf(fr,'%g',inf); %inf 表示读到文件末尾
%画出所读取数据的波形
subplot(321); plot(a);
title('从文本数据文件读取的数据');
fclose(fr); %关闭文件
%打开一个新的数据文件
fw=fopen('c:\temp\bindata.txt','wb');
%用二进制方式把向量 a 中的数据写入该文件
countw=fwrite(fw,a,'real * 4');
fclose(fw);
fb=fopen('c:\temp\bindata.txt','rb');
%用 fread 函数从该二进制数据文件中读取数据
b=fread(fb,countw,'float');
fclose(fb);
%画出所读取数据的波形
subplot(322); plot(b);
title('从二进制文件中读取的数据')
```

输出结果(如图 2.138 所示):



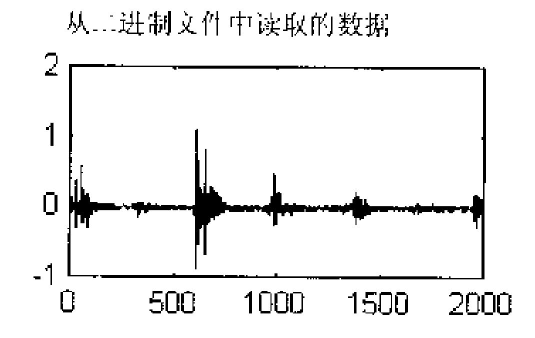


图　2.138

参见: fopen, fclose, fscanf, fprintf, fwrite, ftell, fseek, ferror

9. ftell

功能: 获取文件指针的位置

格式: position=ftell(fid)

说明: 该函数可以获得文件标识为 fid 的文件的指针位置，其返回值是一个非负数，表示从文件开始位置到当前位置的字节数。如果该函数执行错误，则返回−1。

举例: 读取一个文件的位置标识。

```
%打开一个已存在的文件
fid=fopen('mydata.txt');
%读取文件指针的位置
position=ftell(fid)
```

输出结果：

```
position =
    0
```

参见：fopen，fclose，fscanf，fread，fprintf，fwrite，fseek，ferror

10. fseek

功能：设置文件的位置指针

格式：status=fseek(fid,offset,origin)

说明：该函数可以重新设置文件的位置指针，即指针位置按参数 origin 定义的位置移动 offset 个字节(byte)。参数 fid 是由函数 fopen 返回的一个文件标识，参数 offset 定义移动的字节数，具体含义如下：

- 当 offset>0 时，指针位置向文件尾端移动 offset 个字节；
- 当 offset=0 时，指针位置不作移动；
- 当 offset<0 时，指针位置向文件前端移动 offset 个字节。

参数 origin 是一个字符串，它有几种选择：

- 'bof'，表示－1，文件的开始位置；
- 'cof'表示 0，文件当前的位置；
- 'eof'表示 1，文件的末尾。

当函数成功执行时，参数 status 返回 0，否则返回－1。

举例：%打开一个已存在的二进制数据文件

```
fid=fopen('c:\temp\bindata.txt','rb');
%从文件中读取全部数据，读完后，位置指针指向文件末尾处
[a1,count1]=fread(fid,'real * 4');
disp('初始文件中读取数据的个数');
count1
subplot(221); plot(a1); %画出数据的波形图
title('初始文件中读取的数据');
%将文件位置指针从开始处向后移动 1080 个字节(byte)
status=fseek(fid,1080,'bof');
%从移动后的文件中读取数据
[a2,count2]=fread(fid,'real * 4');
disp('前端后移后读取数据的个数');
count2
subplot(222); plot(a2); %画出读取数据的波形图
title('前端后移后读取的数据');
%把位置指针从当前位置(此时指向文件末尾)向前移动 540 个字节(byte)
status=fseek(fid,-540,'cof');
%从移动后的文件中读取数据
[a3,count3]=fread(fid,'real * 4');
```

```
disp('当前位置前移后读取数据的个数');
count3
subplot(223); plot(a3); %画出读取数据的波形图
title('当前位置前移后读取的数据');
%把位置指针从文件末尾向前移动 2160 个字节(byte)
status=fseek(fid,-2160,'eof'); %'eof'等价于-1
%从移动后的文件中读取数据
[a4,count4]=fread(fid,'real*4');
disp('末尾前移后读取数据的个数');
count4
subplot(224); plot(a4); %画出读取数据的波形图
title('末尾前移后读取的数据');
```

输出结果(如图 2.139 所示):

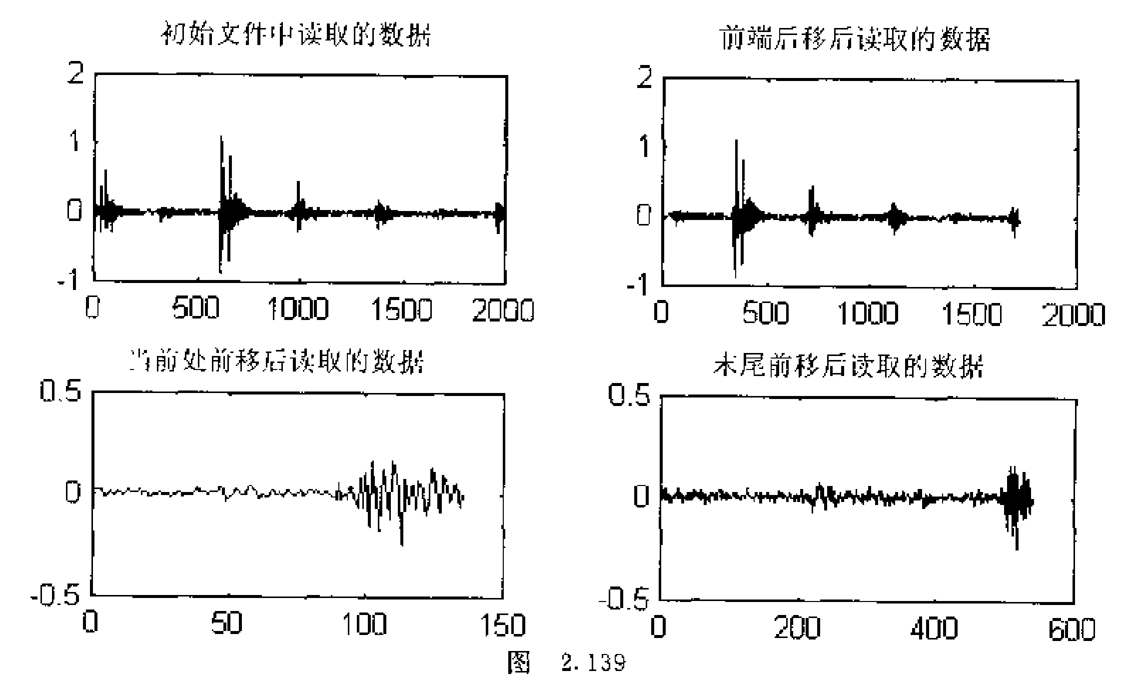


图　2.139

```
初始文件中读取数据的个数
count1 =
        1991

前端后移后读取数据的个数
count2 =
        1721

当前位置前移后读取数据的个数
count3 =
    135
```

末尾前移后读取数据的个数

count4 =

540

从上面的输出结果，我们可以看出：通过文件位置指针的移动，我们可以从二进制数据文件中的任意位置读取数据。

参见：fopen，fclose，fscanf，fread，fprintf，fwrite，ftell，ferror

11. ferror

功能：获取文件 I/O 错误信息

格式：① message=ferror(fid)

② message=ferror(fid,'clear')

③ [message,errnum]=ferror(fid,'clear')

说明：对于格式①，参数 message 返回的是对文件标识为 fid 文件 I/O 的出错信息。参数 fid 是函数 fopen 返回的文件标识。

对于格式②，它在返回出错信息的同时，清除该文件的出错指示符。

对于格式③，在返回 message 出错信息的同时，参数 errnum 还返回最近文件 I/O 的出错状态数。如果文件 I/O 成功，则 errnum 返回 0，message 返回一个空字符串；如果文件 I/O 失败，则 errnum 返回一个非 0 数，message 返回一个包含出错信息的字符串。

举例：读取一个不存在的文件，然后显示出错信息。

```
%正确打开一个数据文件
fid1=fopen('c:\temp\bindata.txt','rb');
%返回上面文件操作信息
[message1, errnum1]=ferror(fid1,'clear')
%进行错误的文件操作
fid2=fopen('c:\temp\magic5.bin','wb');
fwrite(fid2,magic(5),'integer * 4');
%返回上面文件操作信息
[message2,errnum2]=ferror(fid2,'clear')
```

输出结果：

```
message1 =
    ''
errnum1 =
    0


message2 =
Error on flush.
errnum2 =
    -7
```

参见：fopen，fclose，fscanf，fread，fprintf，fwrite，ftell，fseek

12. imread

功能：从图像文件中读取图像数据(二维数据 I/O)

格式：① A=imread(filename,fmt)

② [X,map]=imread(filename,fmt)

③ […]=imread(filename)

④ […]=imread(…,idx) (TIFF only)

⑤ […]=imread(…,ref) (HDF only)

说明：对于格式①，它将图像文件 filename 中的数据读到矩阵 A 中，矩阵 A 的数据类型是 unit8(无符号八位整数)。如果图像文件是一个灰度图像，则 A 是一个二维矩阵；如果图像文件是一个彩色图像(RGB 三色图)，则 A 是一个三维矩阵(m×n×3)。参数 fmt 定义了该文件的图像格式。参数 filename 是一个包含路径名的图像文件名，该文件必须在当前目录或上面指定的目录中。如果 imread 没有发现名为 filename 的文件，则它自动寻找名为 filename. fmt 的文件。

参数 fmt 定义了所读取文件的图像格式，它支持多种图像格式，具体见表 2-15。

表 2-15　参数 fmt 支持的图像格式

格　式	图像文件的类型
'bmp'	Windows 位图(BMP)
'hdf'	分级数据格式(HDF)
'jpg'或'jpeg'	联合图像专家组(JPEG)
'pcx'	Windows 画笔图像(PCX)
'tif'或'tiff'	标签图像格式(TIFF)
'xwd'	X windows dump 图像格式(XWD)

格式②将文件中的索引图像读入矩阵 X，并返回相应的颜色映射矩阵 map。X 的数据类型是 unit8，map 的数据类型是 double。颜色映射的值是[0,1]范围内的一个数。

格式③读取图像文件 filename 的内容，并根据其内容来判断该文件的图像格式。

格式④从一个多图像 TIFF 文件中读取其中的一个图像数据。参数 idx 是一个整数，它是所读取图像在文件中的序号，例如，当 idx=3 时，则函数 imread 读取文件中第三个图像的数据。如果该参数没有显式定义，则读取的是第一个图像的数据。

格式⑤从一个多图像的 HDF 文件中读取其中的一个图像数据。参数 ref 是一个整数，它是所读取图像在文件中的参考号。例如，当 ref=12 时，则函数 imread 读取文件中参考号为 12 的图像数据。在 HDF 文件中，图像的参考号与图像的序号不一定是一致的。如果该参数没有显式定义，则读取的是第一个图像的数据。

表 2-16 总结了函数 imread 可以读取的图像格式。

表 2-16　函数 imread 可以读取的图像格式

格式	变　量
BMP	1 位、4 位、8 位和 24 位的未压缩图像；4 位和 8 位变长度编码(RLE)的图像
HDF	8 位光栅图像数据集，有的带有颜色映射；24 位光栅图像数据集
JPEG	任何基于 JPEG 的图像；经过扩展的 JPEG 图像
PCX	1 位、8 位和 24 位图像
TIFF	任何基于 TIFF 的图像，包括 1 位、8 位和 24 位的未压缩图像；1 位、8 位和 24 位的按比特压缩的图像；1 位按 CCITT 方式压缩的图像
XWD	1 位和 8 位 Z 点图(ZPixmaps)；XY 位图(Bitmaps)；1 位 XY 点图(XYPixmaps)

举例：

例 1：读取 BMP 格式的 lena 灰度图像，并显示。

```
[X,map]=imread('d:\lena.bmp','bmp');
colormap(map);
image(X)
title('显示所读取的 lena 图');
axis square;
```

输出结果(如图 2.140 所示)。

图　2.140

例 2：在一个 TIFF 多图像文件中读取第六个图像的数据。

```
[X,map]=imread('d:\zjb\flower.tif',6);
```

例 3：在一个 HDF 多图像文件中读取第四个图像的数据。

```
[X,map] = imread ('d:\zjb\skull.hdf',info(4),Reference);
```

参见：imwrite，imfinfo

13. imwrite

功能：将图像数据按照一定的图像格式写入图像文件

格式：① imwrite(A,filename,fmt)

② imwrite(X,map,filename,fmt)

③ imwrite(…,filename)

④ imwrite(…,Parameters,Value,…)

说明：格式①将图像矩阵 A 的数据写入名为 filename 的文件。参数 filename 是一个文件名，它可以包含文件的路径；参数 fmt 用于定义所存文件的图像格式。如果 A 是一个灰度图像矩阵或是一个彩色图像(RGB 三色)，图像的数据类型是 unit8，则函数 imwrite 将矩阵 A 中

的原始数据写入文件中。如果 A 中的数据类型是 double，则函数 imwrite 在将数据写入文件之前，用 unit8 的数据格式进行数据转换(转换方法是 round(255 * A))，这样，就将一个位于区间[0,1]的 double 数转化成了一个在[0,255]之间的 8 位整数。

参数 fmt 定义了所写入文件的图像格式，它支持多种图像格式，具体见表 2-15。

格式②将一个索引图像 X 的图像数据写入到文件中，参数 map 是相应的颜色映射矩阵。如果 X 的数据类型是 unit8，则函数 imwrite 是将矩阵的原始数据写入文件；如果 X 的数据类型是 double，则函数 imwrite 将矩阵 X 中的数据进行转换，然后再写到文件中去。它的转换形式是：uint8(X－1).map，转化后的数据类型是 double，这时函数也将 map 中的数据进行转换，转换形式是：unit8(round(255 * map))。

格式③从待写入文件的后缀名中判断出图像的格式之后，再将图像数据写入该文件。所用的文件后缀名必须是一个能识别的、合法的图像后缀名。

格式④中，参数 Parameters 用于定义输出文件中的不同的特征。这个参数目前只支持 HDF、JPEG 和 TIFF 格式的图像文件。

表 2-17 列出了在 HDF 文件中，参数 Parameters 和参数 Values 的几种选项。

表 2-17　在 HDF 文件中，参数 Parameters 和参数 Values 的几种选项

Parameters	Values	默认值
'Compression'	'none'、'rle'、'jpeg'中，三者选一	'rle'
'Quality'	一个在 0 到 100 之间的数：它只在'compression'取'jpeg' 时有用；数值越高表示图像的质量越好(此时图像压缩只引起较少的图像失真)，但图像占用的空间越大	75
'WriteMode'	在'overwrite'和'append'中，二取一	'overwrite'

表 2-18 列出了在 JPEG 文件中，参数 Parameters 和参数 Values 的几种选项。

表 2-18　在 JPEG 文件中，参数 Parameters 和参数 Values 的几种选项

Parameters	Values	默认值
'Quality'	一个在 0 到 100 之间的数：它只在'compression'取'jpeg'时有用；数值越高表示图像的质量越好(此时图像压缩只引起较少的图像失真)，但图像占用的空间越大	75

表 2-19 列出了在 TIFF 文件中，参数 Parameters 和参数 Values 的几种选项。

表 2-19　在 TIFF 文件中，参数 Parameters 和参数 Values 的几种选项

Parameters	Values	默认值
'Compression'	在'none'、'packbits'、'ccitt'中，三者取一；其中'ccitt'只对二进制图像有用	'ccitt'对应二进制图像；'packbits'适用于所有图像
'Description'	可以是任何字符串；它写入图像的 ImageDescription 变量中去，可由 imfinfo 返回得到	空字符串

表 2-20 总结了函数 imwrite 可以写入的图像格式。

表 2-20　函数 imwrite 可以写入的图像格式

格式	变　量
BMP	8 位未压缩图像，带有颜色映射；24 位未压缩图像
HDF	8 位光栅图像数据集，有的带有颜色映射；24 位光栅图像数据集
JPEG	任何基于 JPEG 的图像
PCX	8 位图像
TIFF	任何基于 TIFF 的图像，包括 1 位、8 位和 24 位的未压缩图像；1 位、8 位和 24 位的按比特压缩的图像；1 位按 CCITT 方式压缩的图像
XWD	8 位 Z 点图(ZPixmaps)

举例：

例：将 MATLAB 中的 woman.mat 图像数据存成 PCX 格式的图像文件。

```
%装入图像，X 中包含图像数据
load woman;
colormap(map);
subplot(221); image(X);
title('原始图像');
axis square;
%把 X 中的图像数据写入一个 *.pcx 文件中
imwrite(X,map,'c:\temp\woman1.pcx','pcx');
%从已写入的 *.pcx 文件中读取图像数据
[X1,map1]=imread('c:\temp\woman1.pcx','pcx');
colormap(map1);
subplot(222); image(X1);
title('从 *.pcx 文件中读取的图像');
axis square;
```

输出结果(如图 2.141 所示)：

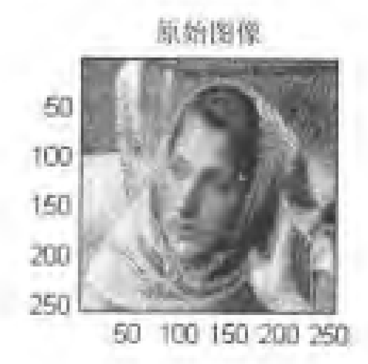


图　2.141

上面的程序运行之后，在 c:\temp 目录下，产生了 woman1.pcx 图像文件。

例 2：将一个图像数据写入 HDF 格式的图像文件中去

load woman;

imwrite(X,map,'c:\temp\woman1.hdf','Compression','none','WriteMode','append')

上面的程序运行之后，在 c:\temp 目录下，产生了 woman1.hdf 图像文件。

参见：imread，imfinfo

14. imfinfo

功能：返回一个图像文件的信息

格式：① info=imfinfo(filename,fmt)

② info=imfinfo(filename)

说明：格式①返回一个图像文件中的各种信息，并将信息放在一个结构体数据中。参数 filename 是图像文件名。参数 fmt 是该文件的图像格式，它可以使图像格式与函数 imread 和 imwrite 中的 fmt 相同，在此不再重复。图像文件必须在当前目录或用户指定的目录中。如果函数 imfinfo 没有发现名为 filename 的图像文件，则它会自动寻找名为 filename.fmt 的文件。

如果参数 filename 是一个 TIFF 或 HDF 文件，并且该文件中包含多个子图像(即多图像文件)，则参数 info 是这个文件信息的结构体数据组，每一个结构体数据包含一个子图像的信息。例如，info(3)包含的是文件中的第三个子图像信息。

对不同的图像格式，参数 info 中的数据块格式不同，下表列出了数据块中通常包括的九个数据项，它们的具体含义见表 2-21。

表 2-21　九个数据项的具体含义

数据项	数 据 项 的 值
Filename	字符串，它包含的是文件名；如果该文件不在当前目录中，则这个字符串还包含文件的路径名
FileModDate	字符串，它包含的是文件最后一次修改的日期
FileSize	整数，以字节(byte)形式指明文件的大小
Format	字符串，它包含图像的格式，由参数 fmt 决定；对 JPEG 或 TIFF 文件，返回的也是三个字母的字符(即 jpg 或 tif)
FormatVersion	字符串或数值，指明该图像格式的版本
Width	整数，以像素点的方式指明图像的宽度
Height	整数，以像素点的方式指明图像的高度
BitDepth	整数，指明每个像素点的比特数(或位数)
ColorType	字符串，指明图像的类型：对 RGB 彩色图像，是'truecolor'，对灰度图像，是'grayscale'，对索引图像，是'indexed'

格式②从图像文件的内容来判断该文件的图像格式。

举例：

例：读取一个 BMP 文件的格式信息。

info=imfinfo('c:\temp\lake. bmp')

输出结果：

```
info =
             Filename: 'c:\temp\lake. bmp'
          FileModDate: '07 - Oct - 1998 22:26:22'
             FileSize: 253494
               Format: 'bmp'
        FormatVersion: 'Version 3 (Microsoft Windows 3. x)'
                Width: 352
               Height: 240
             BitDepth: 24
            ColorType: 'truecolor'
      FormatSignature: 'BM'
   NumColormapEntries: 0
             Colormap: []
              RedMask: []
            GreenMask: []
             BlueMask: []
      ImageDataOffset: 54
     BitmapHeaderSize: 40
            NumPlanes: 1
      CompressionType: 'none'
           BitmapSize: 253440
       HorzResolution: 0
       VertResolution: 0
        NumColorsUsed: 0
   NumImportantColors: 0
```

参见：imread，imwrite

第 3 章　小波分析的应用技术

随着小波理论的日益成熟，人们对小波分析的实际应用越来越重视，它已经广泛地应用于信号处理、图像处理、量子场论、地震勘探、话音识别与合成、音乐、雷达、CT 成像、彩色复印、流体湍流、天体识别、机器视觉、机械故障诊断与监控、分形以及数字电视等科技领域。在本章，主要介绍如何利用小波分析函数处理一些实际的工程问题。

在上一章，我们知道，MATLAB 所提供的小波分析工具有两种：第一种提供的是命令行形式的函数，用户可通过这些函数，根据实际分析的需要，在调试状态下，编写自己的 MATLAB 程序。这种方式虽不直观，但它可以按照用户自己的思维，编写出功能强大的 MATLAB 程序，完成各种信号的小波分析。第二种提供的是图形用户接口(Graphic User Interface：GUI)工具，这种方式简单直观，不需要进行复杂的编程，并且将计算结果直接以图形方式显示出来，用户可以立即评价自己分析的结果是否正确，但是 GUI 方式的处理模式比较固定，且它所提供的小波函数种类较少，在进行复杂的信号分析时，有些功能无法实现。从思维角度来讲，利用 MATLAB 提供的命令行形式的函数编程，可以领会小波分析中的许多细节部分，因此在本章中，我们只讲解第一种方式的应用。

3.1　一维小波分析的应用

首先，我们将用于一维小波图像分析的主要函数作一个简要介绍，这些函数在第 2 章中已做过详细说明，在此，为了方便读者的使用而作一个归纳总结，具体每个函数的用法，请参阅第 2 章的有关内容。用于一维信号分析的函数主要有：

(1) 小波分解函数

表 3-1　一维小波分解函数

函数名	功　能
cwt	一维连续小波变换
dwt	单尺度一维离散小波变换
dwtper	单尺度一维离散小波变换(周期性)
wavedec	多尺度一维小波分解(一维多分辨分析函数)

(2) 小波重建函数

表 3-2　一维小波重建函数

函数名	功　　能
idwt	单尺度一维离散小波逆变换
idwtper	单尺度一维离散小波重构(周期性)
waverec	多尺度一维小波重构
upwlev	单尺度一维小波分解的重构
wrcoef	对一维小波系数进行单支重构
upcoef	一维系数的直接小波重构

(3) 分解结构应用函数

表 3-3　一维小波分解结构应用函数

函数名	功　　能
detcoef	提取一维小波变换高频系数
appcoef	提取一维小波变换低频系数

(4) 噪声函数

表 3-4　噪 声 函 数

函数名	功　　能
wnoise	产生含噪声的测试函数数据
wnoisest	估计一维小波的系数的标准偏差

(5) 消噪和压缩函数

表 3-5　一维小波消噪和压缩函数

函数名	功　　能
thselect	信号消噪的阈值选择
wthresh	进行软阈值或硬阈值处理
wthcoef	一维信号的小波系数阈值处理
wden	用小波进行一维信号的自动消噪
ddencmp	获取在消噪或压缩过程中的默认值阈值(软或硬)、熵标准
wdencmp	用小波进行信号的消噪和压缩

3.1.1　小波分析的一些数学计算

在这里，我们以小波分析这一数学工具处理一些数学问题，从某种意义上讲，这种应用是帮助读者对小波分析理论本身有进一步的理解。

例 3.1：对于一给定的正弦信号 $s(i) = \sin(i \times \pi/100 + \pi/4)$，$i = 0,1,\cdots,199$，请利

用多分辨分析对该信号进行分解与重构。

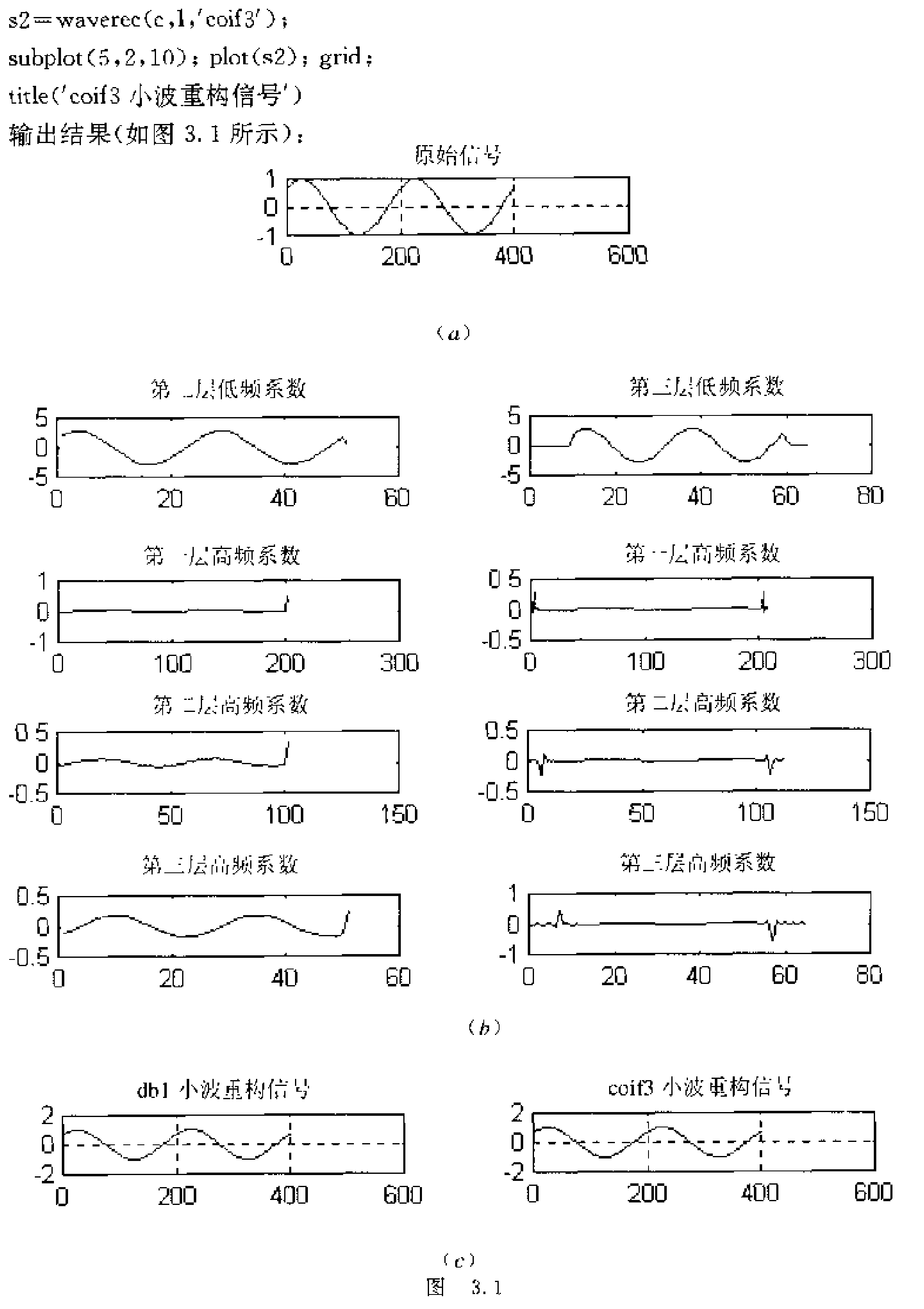
解：该问题可以说是一个纯粹的数学问题，通过对该问题的讲解，我们可以加深对小波分析中的多分辨分析的理解，即如何对信号进行多层分解与重构。

在这里，我们分别选用 db1 和 coif3 小波对该正弦信号进行三层多分辨分析，处理过程可编程如下：

```
t=0 : pi/100 : 4 * pi;
s=sin(t+pi/4);
subplot(532); plot(s);
title('原始信号');
[c,l]=wavedec(s,3,'db1'); grid;
ca3=appcoef(c,l,'db1',3); %提取小波分解的低频系数
cd3=detcoef(c,l,3); %提取第三层的高频系数
cd2=detcoef(c,l,2); %提取第二层的高频系数
cd1=detcoef(c,l,1); %提取第一层的高频系数
figure(2);
subplot(521); plot(ca3);
title('第三层低频系数');
subplot(523); plot(cd1);
title('第一层高频系数');
subplot(525); plot(cd2);
title('第二层高频系数');
subplot(527); plot(cd3);
title('第三层高频系数');
s1=waverec(c,l,'db1');
subplot(529); plot(s1); grid;
title('db1 小波重构信号');
[c,l]=wavedec(s,3,'coif3');
ca3=appcoef(c,l,'coif3',3); %提取小波分解的低频系数
cd3=detcoef(c,l,3); %提取第三层的高频系数
cd2=detcoef(c,l,2); %提取第二层的高频系数
cd1=detcoef(c,l,1); %提取第一层的高频系数
subplot(522); plot(ca3);
title('第三层低频系数');
subplot(524); plot(cd1);
title('第一层高频系数');
subplot(526); plot(cd2);
title('第二层高频系数');
subplot(528); plot(cd3);
title('第三层高频系数');
```

```
s2=waverec(c,l,'coif3');
subplot(5,2,10); plot(s2); grid;
title('coif3 小波重构信号')
```

输出结果(如图 3.1 所示):



(a)

(b)

(c)

图　3.1

例 3.2　给定一信号(信号文件名为 leleccum.mat),请用 db1 小波对信号分别进行单尺度和三尺度分解,求出各次分解的低频系数和高频系数,并分别对低频系数、高频系数以及低高频系数进行重构。

解:该问题是运用小波分析进行信号单、多尺度分解与重构方法的一次全面的讨论。通过对该问题的分析,主要让读者清楚如何用小波分析对信号进行单尺度、多尺度分解以

及对信号进行部分重构和全面重构。

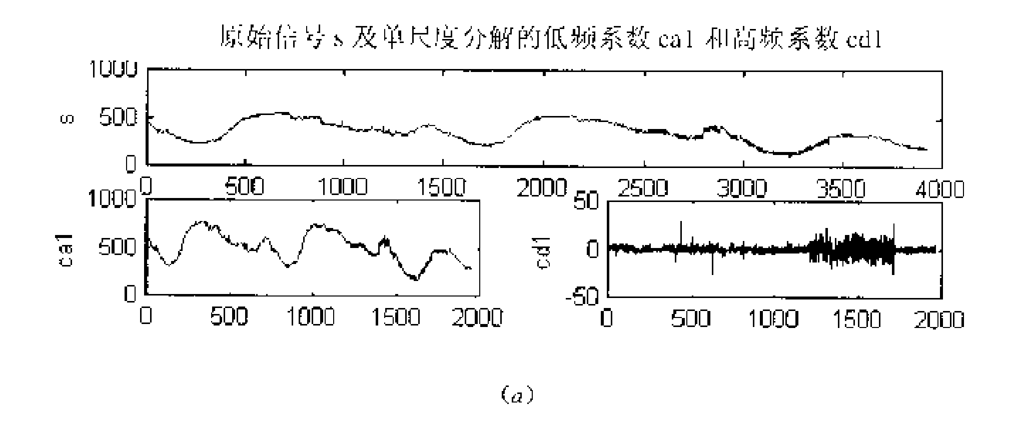
程序清单：

```
% 装载一原始的一维信号
load leleccum; s = leleccum(1 : 3920);
ls = length(s);
%画出原始波形图
figure(1);
subplot(511); plot(s);
Ylabel('s');
title('原始信号 s 及单尺度分解的低频系数 ca1 和高频系数 cd1');
% 用小波 db1 进行单尺度一维分解
[ca1,cd1] = dwt(s,'db1');
%画出分解后的低频系数 ca1 和高频系数 cd1
subplot(523); plot(ca1);
Ylabel('ca1');
subplot(524); plot(cd1);
Ylabel('cd1');
% 分别对低频系数 ca1 和高频系数 cd1 进行重构
a1 = upcoef('a',ca1,'db1',1,ls);
d1 = upcoef('d',cd1,'db1',1,ls);
%分别画出重构后低频部分和高频部分的波形图
figure(2);
subplot(511); plot(a1);
Ylabel('a1');
title('单尺度分解的低频重构信号、高频重构信号及合成重构信号')
subplot(512); plot(d1);
Ylabel('d1');
% 画出 a1 + d1 的波形图，即对 s 的分解系数直接进行重构
a0 = idwt(ca1,cd1,'db1',ls);
subplot(513); plot(a0);
Ylabel('a1+d1');
% 用 db1 小波对信号进行三层分解
[c,l] = wavedec(s,3,'db1');
% 从小波分解结构[c,l]中提取低频系数
ca3 = appcoef(c,l,'db1',3);
% 从小波分解结构[c,l]中提取第 1，2，3 层的高频系数
cd3 = detcoef(c,l,3);
cd2 = detcoef(c,l,2);
cd1 = detcoef(c,l,1);
```

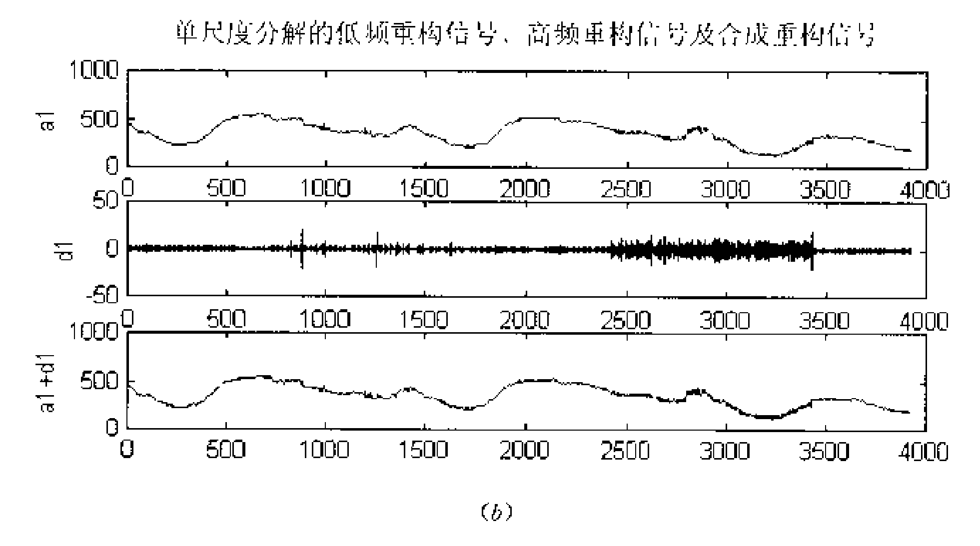
```
%分别画出原始信号、低频系数和高频系数的波形
figure(3);
subplot(511); plot(s);
Ylabel('s');
title('原始信号及三层分解的各层分解系数');
subplot(589); plot(ca3);
Ylabel('ca3');
subplot(5,8,17); plot(cd3);
Ylabel('cd3');
subplot(5,4,13); plot(cd2);
Ylabel('cd2');
subplot(5,2,9); plot(cd1);
Ylabel('cd1');
% 对第 3 层的低频系数进行重构
a3 = wrcoef('a',c,l,'db1',3);
% 从小波分解结构[c,l]中提取第 1，2，3 层的高频系数进行重构
d3 = wrcoef('d',c,l,'db1',3);
d2 = wrcoef('d',c,l,'db1',2);
d1 = wrcoef('d',c,l,'db1',1);
% 画出各层系数重构后的波形图
figure(4);
subplot(511); plot(ca3);
Ylabel('ca3');
title('各层分解系数的重构图及合成重构图 a0');
subplot(512); plot(cd3);
Ylabel('cd3');
subplot(513); plot(cd2);
Ylabel('cd2');
subplot(514); plot(cd1);
Ylabel('cd1');
% 对小波分解结构[c,l]进行重构
a0 = waverec(c,l,'db1');
subplot(515); plot(a0);
Ylabel('a0');
```

输出结果(如图 3.2 所示)。

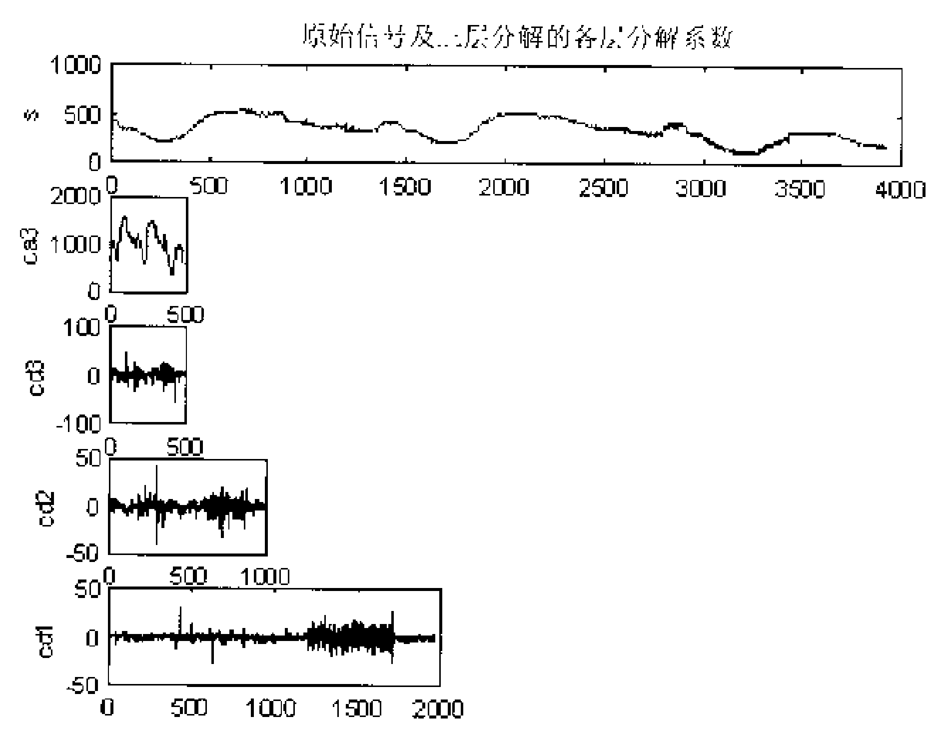
从上面的分析，我们可以看出：小波分析可以把信号的不同频率区域分开。


原始信号 s 及单尺度分解的低频系数 ca1 和高频系数 cd1
s
ca1
cd1

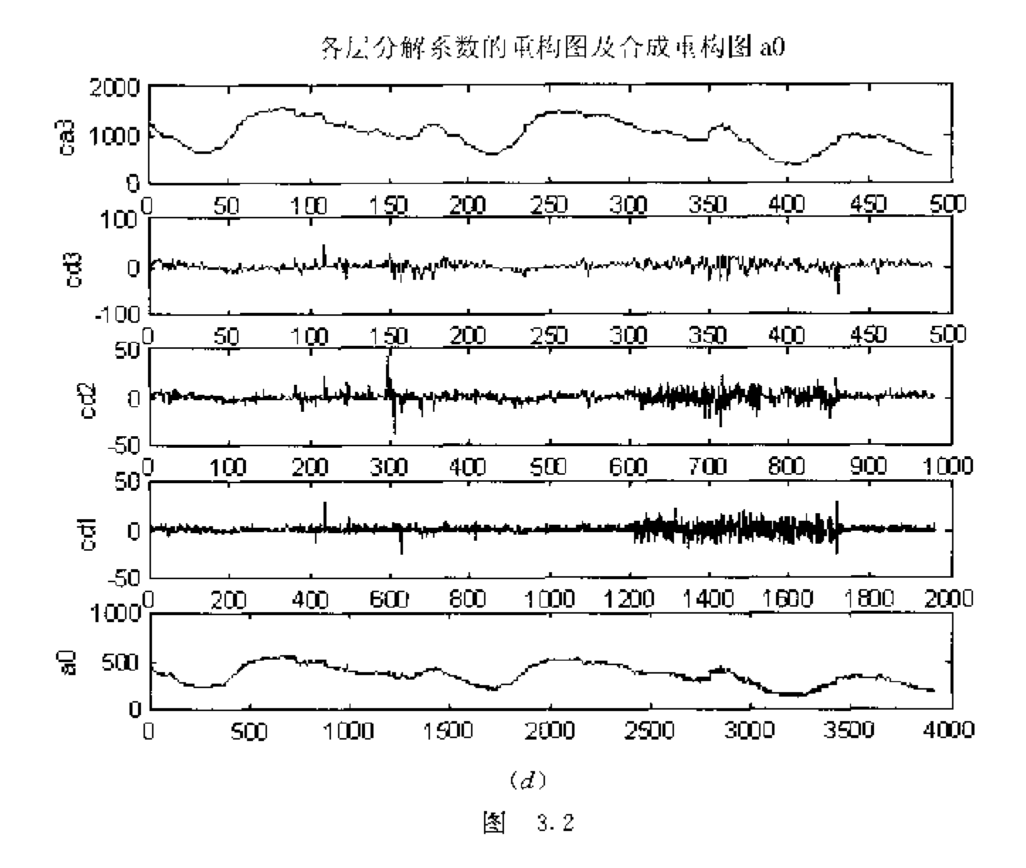
(a)


单尺度分解的低频重构信号、高频重构信号及合成重构信号
a1
d1
a1+d1

(b)


原始信号及三层分解的各层分解系数
s
ca3
cd3
cd2
cd1

(c)



(*d*)

图 3.2

3.1.2 小波函数中的过零点分析

下面我们先简单地介绍一下过零点及相关的一些概念。

定义 3.1 在某一尺度 a_0 下，如果存在一点 (a_0,b_0) 使得 $\frac{\partial W_f(a_0,b_0)}{\partial b}=0$，则称点 (a_0,b_0) 是局部极值点，且 $\frac{\partial W_f(a_0,b)}{\partial b}$ 在 $b=b_0$ 上有一**过零点**。如果对 b_0 的某一邻域内的任意点 b，有 $|W_f(a_0,b)|\leqslant|W_f(a_0,b_0)|$，则称 (a_0,b_0) 为小波变换的**模极大值点**。尺度空间 (a,b) 中所有的模极大值点的连线称为**极大值线**。

由定义 3.1 我们看出小波变换模极大值点 (a_0,b_0) 在点 b_0 的右邻域和左邻域都是严格局部最大的。关于模极值与函数的奇异性有下面的定理：

定理 3.1 设 n 为一严格的整数，Ψ 为 n 阶消失矩、n 次连续可微和具有紧支集的小波，$f(t)\in L^1(c,d)$（$[c,d]$ 为某一实数区间），若存在尺度 $a_0>0$，使得 $\forall\ a<a_0$，$t\in(c,d)$，$|W_f(a,b)|$ 没有局部极大值点，则在区间 $(c+\varepsilon,d-\varepsilon)$ 上是一致 Lipschitz α（ε 为一任意小的正数）。

3.1.3 信号奇异性检测

信号中的奇异点及不规则的突变部分经常带有比较重要的信息，它是信号重要的特征之一。比如，在故障诊断（特别是机械故障诊断）中，故障通常表现为输出信号发生突变，因而对突变点的检测在故障诊断中有着非常重要的意义。长期以来，傅里叶变换是研究函数奇异性的主要工具，其方法是研究函数在傅里叶变换域的衰减以推断函数是否具有奇异性及奇异性的大小。但傅里叶变换缺乏空间局部性，它只能确定一个函数奇异性的整体性

质，而难以确定奇异点在空间的位置及分布情况。我们知道，小波变换具有空间局部化性质，因此，利用小波变换来分析信号的奇异性及奇异性位置和奇异度的大小是比较有效的。

通常情况下，信号奇异性分两种情况：一种是信号在某一个时刻内，其幅值发生突变，引起信号的非连续，幅值的突变处是**第一种类型的间断点**；另一种是信号外观上很光滑，幅值没有突变，但是，信号的一阶微分有突变产生，且一阶微分是不连续的，称为**第二种类型的间断点**。

通常，用李普西兹指数(Lipschitz)来描述函数的局部奇异性。下面我们给出一个描述信号奇异度的一般定义。

定义 3.2　设 n 是一非负整数，$n<\alpha\leqslant n+1$，如果存在着两个常数 A 和 $h_0(>0)$，及 n 次多项式 $P_n(h)$，使得对任意的 $h\leqslant h_0$，均有

$$|f(x_0+h)-P_n(h)|\leqslant A|h|^\alpha$$

则说 $f(x)$ 在点 x_0 为 Lipschitz α。

如果上式对所有 $x_0\in(a,b)$ 均成立，且 $x_0+h\in(a,b)$，称 $f(x)$ 在 (a,b) 上是一致 Lipschitz α。

显然，$f(x)$ 在 x_0 点的 Lipschitz α 刻画了函数在该点的正则性，称为函数 $f(x)$ 在 x_0 点是 Lipschitz α。Lipschitz α 指数越大，函数越光滑；函数在一点连续、可微，则在该点的 Lipschitz α 指数为 1。在一点可导，而导数有界但不连续时，Lipschitz α 指数仍为 1；如果 $f(x)$ 在 x_0 的 Lipschitz $\alpha<1$，则称函数在 x_0 点是奇异的。一个在 x_0 点不连续但有界的函数，该点的 Lipschitz α 指数为 0。

在利用小波分析这种局部奇异性时，小波系数取决于 $f(x)$ 在 x_0 点的邻域内的特性及小波变换所选取的尺度。在小波变换中，局部奇异性可定义为：

定义 3.3　设 $f(x)\in L^2(R)$，若 $f(x)$ 对 $\forall\ x\in\delta x_0$，小波 $\Psi(x)$ 满足实且连续可微、并具有 n 阶消失矩(n 为正整数)，有：

$$|Wf(s,x)|\leqslant Ks^\alpha \qquad K\text{ 为常数}$$

则称 α 为 x_0 点的奇异性指数(也称 Lipschitz 指数)。

定义 3.4　对 $\forall\ x\in\delta x_0$，有 $|Wf(s,x)|\leqslant|Wf(x,x_0)|$，则称 x_0 为小波变换在尺度 s 下的局部极值点。

1. 检测第一种类型的间断点

在这个例子中，介绍了在用小波分析来检测第一类间断点情况下，信号幅值变化的准确时间，即间断点的准确位置。在这个例子中，信号的不连续是由于低频特征的正弦信号在后半部分中，突然有中高频特征的正弦信号加入了。分析的目的是将中高频特征的正弦信号加入的时间点检测出来。

例 3.3　对一个给定的含有突变点的信号(信号的文件名为 freqbrk.mat)，请利用小波分析对信号突变点的时机进行检测。

解：该问题的处理可按如下程序进行。

```
load freqbrk；%装入要分析的信号
s=freqbrk；
```

```
ls=length(s);
[c,l]=wavedec(s,6,'db5');        %用 db5 小波分解信号到第六层
subplot(8,1,1); plot(s);
title('用 db5 小波分解六层:s=a6+d6+d5+d4+d3+d2+d1'); Ylabel('s');
%对分解结构[c,l]中的第六层低频部分进行重构
a6=wrcoef('a',c,l,'db5',6);
subplot(8,1,2); plot(a6); Ylabel('a6');
%对分解结构[c,l]中的各层高频部分进行重构
for i=1:6
    decmp=wrcoef('d',c,l,'db5',7-i);
    subplot(8,1,i+2);
    plot(decmp);
    Ylabel(['d',num2str(7-i)]);
end
```

输出结果(如图 3.3 所示):

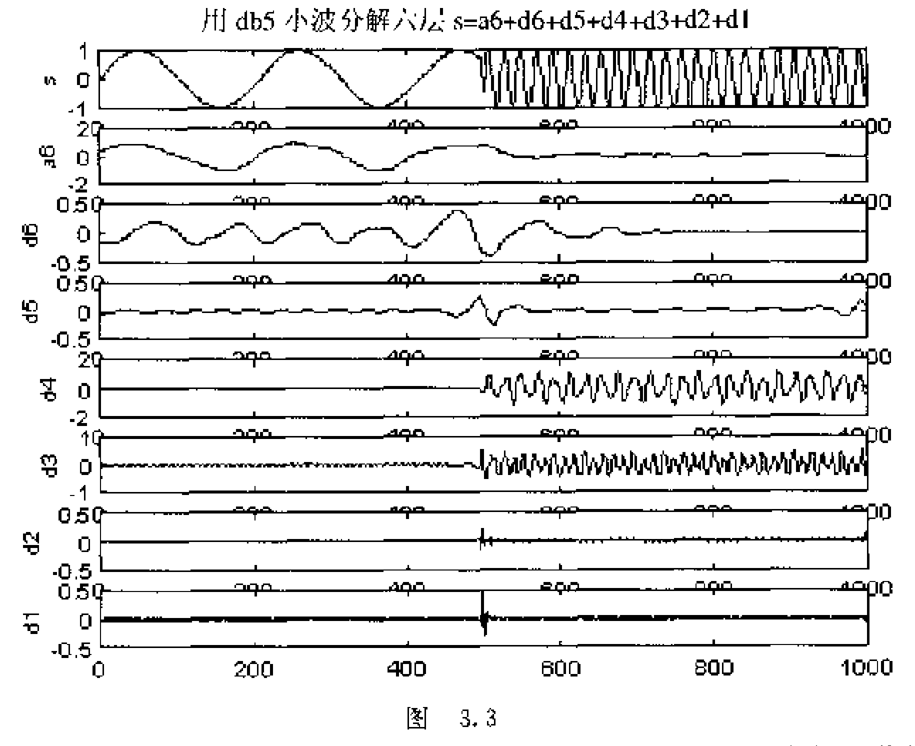


图 3.3

我们看到，在该信号的小波分解中，第一层(D1)和第二层(D2)的高频部分将信号的不连续点显示得相当明显，因为信号的断裂部分包含的是高频部分。这里需要说明的是，如果我们只想辨别出信号的不连续点，用 db1 小波比 db5 小波效果更好。由上图可以看出，信号不连续点的时域定位非常精确，即该点在时域中($t\approx500$)一个非常小的范围之内。这种情况一般是在小波分解的第一层和第二层高频中判断。

这个例子有力地说明了小波分析比传统的傅立叶分析有更大的优越性。如果这种信号用傅立叶分析方法进行，我们在频域中是无法检测出信号在时域中的突变点的，而在小波

分析中，这种突变点的特征就表现得相当明显。

在信号处理中，信号中含有噪声是一种相当普遍的情况，而噪声的存在增加了辨别信号不连续点的复杂性。一般说来，如果信号用小波分解的第一层能够估计出噪声的大体位置，则信号的断裂点(频率变化点)就能够在小波分解的更深的层次上显示出来。

下面给出用傅立叶变换和 db1 小波变换实现的对该信号在频域内的表现形式，从比较中可以更好地领会傅立叶变换在处理时域上有突变的信号的不足之处，以及用不同的小波变换，对信号分析的差别。

程序清单：

```
%信号的傅立叶变换
load freqbrk；%装入要分析的信号
s=freqbrk；
ls=lengtb(s)；
subplot(6,1,1)；plot(s)；title('原始信号的时域图')；
%对信号 s 进行 FFT 变换
fs=fft(s,1024)；%在 s 信号中取 1024 个点，倘若 s 中不够长，则后面补零
fs=abs(fs)；%将 FFT 后的复数用 abs 求其模的大小，返回的值是复数的模
subplot(6,1,2)；plot(fs)；
Ylabel('FFT')；grid；
%信号用 db1 小波分解到第三层后的频域特性
[c,l]=wavedec(s,3,'db1')；      %用 db1 小波分解信号到第三层
%对分解结构[c,l]的第三层低频部分进行重构
a3=wrcoef('a',c,l,'db1',3)；
subplot(6,1,3)；plot(a3)；Ylabel('a3')；
%对分解结构[c,l]中的各层高频部分进行重构
for i=1：3
    decmp=wrcoef('d',c,l,'db3',4-i)；
    subplot(6,1,i+3)；
    plot(decmp)；
    Ylabel(['d',int2str(4-i)])；
end
```

输出结果(如图 3.4 所示)。

在图中可以看出，由于傅立叶变换将信号变换成纯频域中的信号，使它不具有时间分辨的能力，故对信号在时域中的突变点根本无法检测出来。而 db1 小波分解后的信号，则可以很明显地辨别出该断裂点。

例 3.4　某一正在工作的系统，其正常工作时，输出点的采样信号应为一蠕变信号，当系统出现故障时，输出信号将会出现一突变信号(主要表现在幅度和频率的突变)。现给出该系统从正常到出现故障的一采样序列，请利用小波分析方法分析故障出现的时间点。

解：该问题是一个检测突变点(或不连续点)的问题，利用小波分析可以精确地检测出信号突变的时间点。

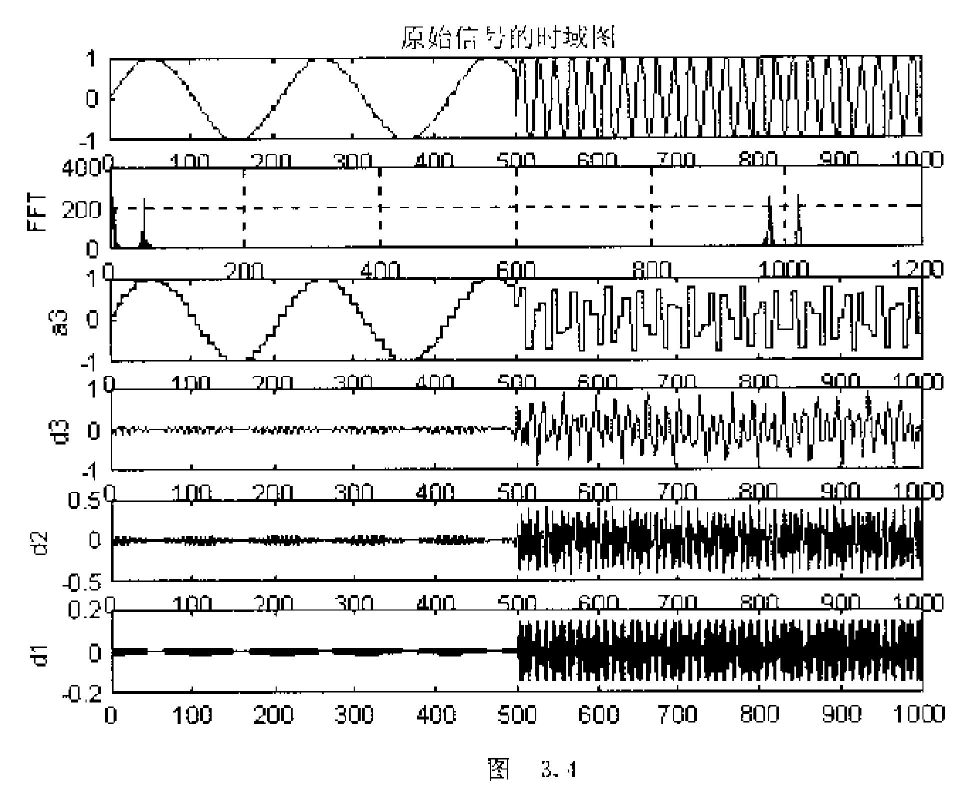


图 3.4

利用小波分析检测信号突变点的一般方法是：对信号进行多尺度分析，在信号出现突变时，其小波变换后的系数具有模量极大值，因而可以通过对模量极大值点的检测来确定故障发生的时间点。

程序清单：

```
t=0:pi/125:4*pi;
s1=sin(t); %设置一正常信号
s2=sin(10*t); %设置一故障信号，表现在频率的突变
    s3=sin(t); %设置一正常信号
    s=[s1,s2,s3]; %整个信号
subplot(421); plot(s);
title('原始信号');
Ylabel('s');
[c,l]=wavedec(s,6,'db3');   %采用db3小波并对信号进行六层分解
apcmp=wrcoef('a',c,l,'db3',6);
subplot(422); plot(apcmp);
Ylabel('ca6');
for i=1:6
    decmp=wrcoef('d',c,l,'db3',7-i);
    subplot(4,2,i+2);
    plot(decmp);
    Ylabel(['d',num2str(7-i)]);
```

```
end
```

输出结果(如图 3.5 所示)：

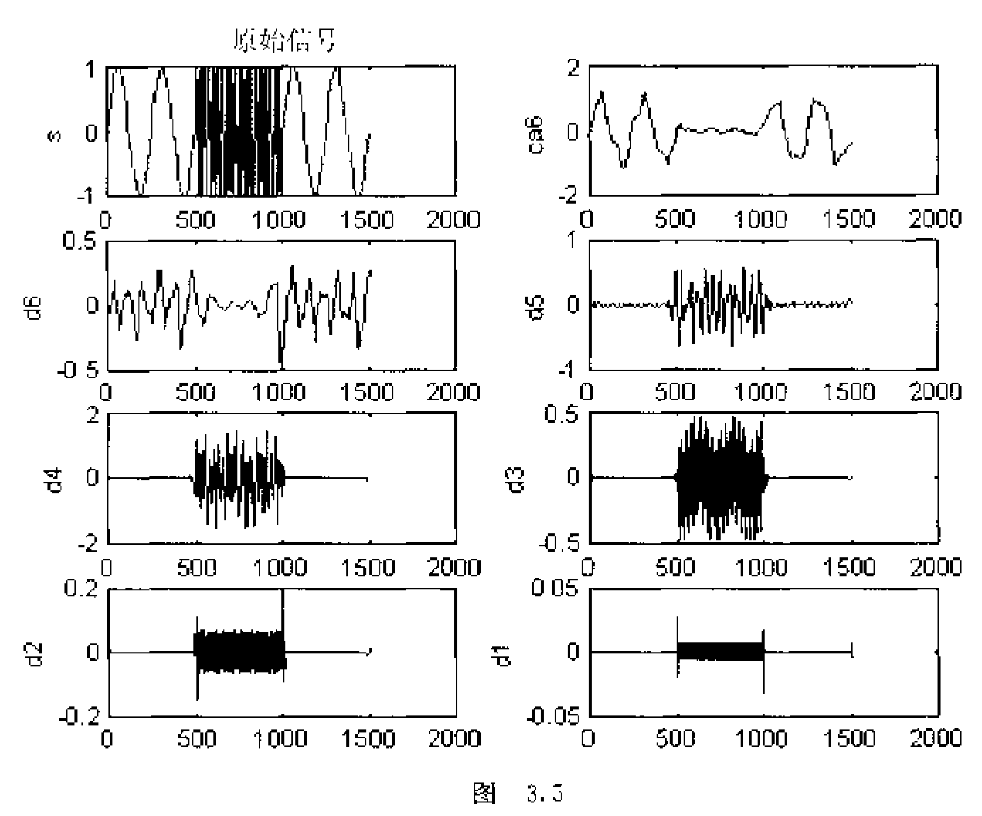


图　3.5

从上图的小波分解的层系数可以明显看出，在 t=500 时，系统工作出现了异常情况，在 t=1000 时，系统工作又恢复了正常。

这是一个利用小波分析检测信号突变点的典型实例。从该例可以看出，小波分析在检测信号突变点(奇异点)上具有比傅里叶变换无法比拟的优越性，利用小波分析可以精确地检测出信号突变时间点。

2. 检测第二种类型的间断点

例 3.5　对某一给定的信号，它是由两个独立的满足指数方程的信号连接起来的．请利用小波分析来检测出第二类间断点的准确位置。

解：这个例子中的信号，在外观上是很光滑的曲线，但是，该信号具有一阶微分且突变。分析的目的是将第二类间断点寻找出来。

程序清单：

```
t=1:0.01:2;
s1=exp(t);
s2=exp(4*t);
s=[s1,s2]; %设置由不同的指数函数组成的信号
subplot(6,1,1); plot(s); title('原始信号');
ds=diff(s); %计算信号的一阶微分
%显示信号的一阶微分结果
```

```
subplot(6,1,2); plot(ds);
Ylabel('s 微分');
[c,l]=wavedec(s,2,'db1'); %用 db1 小波分解信号到第二层
%对分解结构[c,l]中的第二层低频部分进行重构
a2=wrcoef('a',c,l,'db1',2);
%显示重构结果
subplot(6,1,3); plot(a2);
Ylabel('a2');
%对分解结构[c,l]中的各层高频部分进行重构并显示结果
d2=wrcoef('d',c,l,'db1',2);
subplot(6,1,4); plot(d2);
Ylabel('d2');
d1=wrcoef('d',c,l,'db1',1);
subplot(6,1,5); plot(d1);
Ylabel('d1');
```

输出结果(如图 3.6 所示):

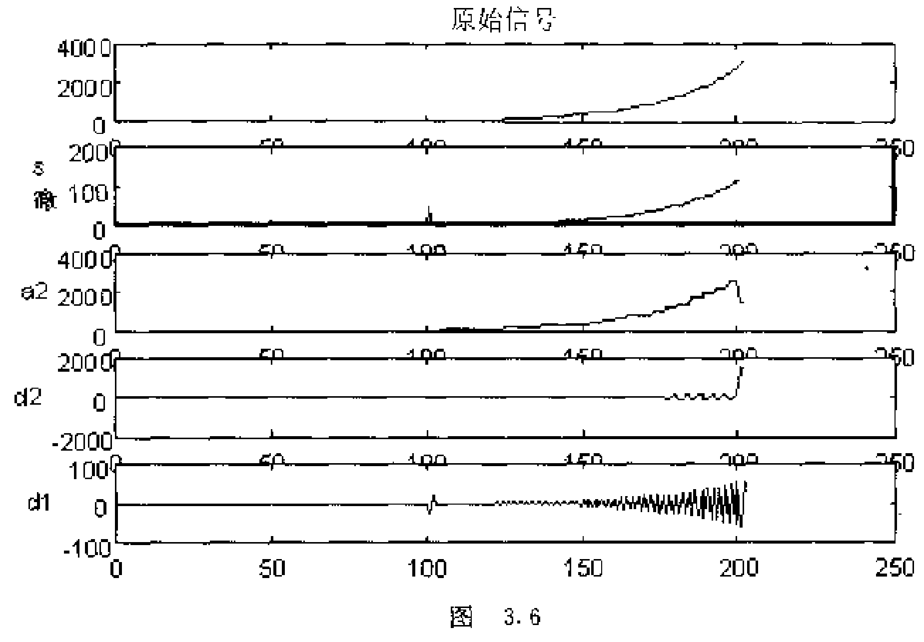


图 3.6

我们看到，该信号的一阶微分曲线在 t=100 点处，有明显的不连续。将该信号进行小波分解后，第一层的高频部分 d1 将信号的不连续点显示得相当明显，这个断裂点在信号的中部发生，在其它地方可以忽略。由上图可以看出，利用小波分析进行信号的不连续点的定位非常精确。像这种间断点的定位，一般来说，是在小波分解的第一层和第二层高频部分中进行判断的。

注意：在选择小波的过程中，正则性是一条很重要的规则，在这里我们选择的是 db1 小波，这种小波正则性很好，如果选择 db4 小波，会发现在 t=100 点处，高频部分的值几乎为 0，检测不出信号的不连续点(第二类间断点)。

为了检测出信号中的奇异点，所选择的小波必须很正则(有规则)，这时的小波可实现

一个更长的冲击响应滤波器。

3.1.4　小波分析用于信号消噪处理

1. 噪声信号的小波分析特性

运用小波分析进行一维信号消噪处理是小波分析的一个重要应用之一，下面我们将其消噪的基本原理作一个简要的说明。

一个含噪声的一维信号的模型可以表示成如下的形式：

$$s(i) = f(i) + \sigma \cdot e(i),\ i = 0,\ \cdots,\ n-1$$

其中，$f(i)$为真实信号，$e(i)$为噪声，$s(i)$为含噪声的信号。

在这里，我们以一个最简单的噪声模型加以说明，即认为$e(i)$为高斯白噪声 $N(0,1)$，噪声级(noise level)为 1。在实际的工程中，有用信号通常表现为低频信号或是一些比较平稳的信号，而噪声信号则通常表现为高频信号。所以消噪过程可按如下方法进行处理：首先对信号进行小波分解(如进行三层分解，分解过程如图 3.7 所示)，则噪声部分通常包含在 cD1,cD2,cD3 中，因而，可以以门限阈值等形式对小波系数进行处理，然后对信号进行重构即可以达到消噪的目的。对信号 $s(i)$消噪的目的就是要抑制信号中的噪声部分，从而在 $s(i)$中恢复出真实信号 $f(i)$。

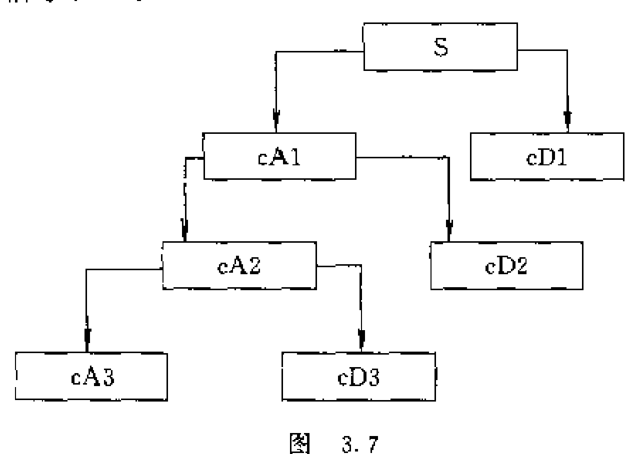


图　3.7

一般说来，一维信号的消噪过程可分为三个步骤进行：

(1) 一维信号的小波分解。选择一个小波并确定一个小波分解的层次 N，然后对信号 s 进行 N 层小波分解。

(2) 小波分解高频系数的阈值量化。对第 1 到第 N 层的每一层高频系数，选择一个阈值进行软阈值量化处理。

(3) 一维小波的重构。根据小波分解的第 N 层的低频系数和经过量化处理后的第 1 层到第 N 层的高频系数，进行一维信号的小波重构。

在这三个步骤之中，最关键的就是如何选取阈值和如何进行阈值的量化，从某种程度上说，它直接关系到信号消噪的质量。

在上面的说明过程中，我们是以最简单的信号模型对信号消噪进行说明，

$$s(i) = f(i) + \sigma \cdot e(i),\ i = 0,\ \cdots,\ n-1$$

在实际中，这种最基本的模型是不能直接运用的，为了求得模型的标准偏差，需要对

信号消噪的主要函数 wden 的输入参数进行设置。该函数最简单的用法是：

sd＝wden(s,tptr,sorh,scal,n,wav)

它返回的是经过对原始信号 s 进行消噪处理后的信号 sd。另外，sorh 指定软阈值(sorh＝'s')或硬阈值(sorh＝'h')的选择，tptr 指定阈值选取的规则，n 为小波分解的层数，wav 指定分解时所用的小波，参数 scal 是阈值尺度改变的比例，它有以下三种选择，具体说明见表 3－6。

表 3－6 参数 scal 的选项

scal 的选项	相应的模式
'one'	基本模式
'sln'	未知尺度的基本模式
'mln'	非白噪声的基本模式

一般说来，可以忽略噪声的层次。在小波分解的高频层 cD1(最精细的尺度)中，主要是噪声的系数，它的标准偏差等于 σ；绝对标准偏差比较稳定，其估计值总是 σ。这种稳定的估计值在信号分析中很重要，主要有两个原因：第一个是如果第一层的系数中含有信号 f 的高频信息，并且信号本身是很规则的，则这种高频信息在几个高频层中都能集中显示出来；第二个原因是可以避免信号本身的截短效应，这种截短效应是在计算信号的边缘时，产生的无用信息。当认为噪声 e 是一个非白噪声时，必须在每个不同的小波分解层次上估计噪声的层次，并由此来变换阈值尺度，即根据小波分解的层次 lev，来估计 σ_{lev}，这种估计算法在函数 wnoise 中实现，它是通过直接计算信号 s 的小波分解的结构来实现的。scal＝'sln'时，只根据第一层的小波分解系数来估计噪声的层次，并且只进行一次估计。以此来变换阈值的尺度。scal＝'mln'时，在每个不同的小波分解层次上都估计噪声的层次，并且变换阈值尺度。

另外一种更普遍的函数是 wdencmp，它可以直接对一维信号或二维信号进行消噪或压缩处理，方法也是通过对小波分解系数进行阈值量化来实现的。它可以让用户选择自己的阈值量化方案，其使用方式是：

xd＝wdencmp(opt,wav,n,thr,sorh,keepapp)；

- opt＝'gbl'，并且 thr 是一个正的实数，则阈值是全局阈值。
- opt＝'lvd'，并且 thr 是向量，则阈值是在各层大小不同的数值。
- keepapp＝1，将小波分解的低频系数不作任何处理。
- keepapp＝0，对小波分解的低频系数也进行阈值量化处理。
- x 是待处理的输入信号。
- xd 是处理后的信号。
- 其余输入参数与 wden 命令函数的相同。

2. 噪声在小波分解下的特性

在这里，我们将噪声看成是一个普通的信号，并对它进行分析，那么有三个主要的特征需要注意，即相关性、频谱和频率分布。

总体上来说，一个一维离散的信号，它的高频部分影响的是小波分解的高频第一层，低频部分影响的是小波分解的最深层及其低频层。如果对一个只是由白噪声组成的信号进行小波分解，则可以看出：高频系数的幅值随着分解层次的增加而很快地衰减，并且，高频系数的方差也很快地衰减。当通过滤波器将有色噪声引入后，该信号就不是白噪声了。对噪声用小波分解的系数仍用 $C(j,k)$ 表示，其中，j 代表小波尺度，k 代表时间，我们可以对噪声信号引入一些常用的属性：

（1）如果所分析的信号 s 是一个平稳、零均值的白噪声，则其小波分解系数是不相关的。

（2）如果 s 是一个高斯噪声，则其小波分解系数是独立的，也是高斯分布的。

（3）如果 s 是一个有色、平稳、零均值的高斯噪声序列，则其小波分解系数也是高斯序列。对每一个分解尺度 j，其系数是一个有色、平稳的序列。我们感兴趣的是如何选择小波，对分解系数解相关(de - correlate)，这个问题在目前还没有很好地解决。进一步需要指出的是，即使一个小波确实存在，它对噪声的解相关性也取决于信号噪声的有色性。为了用小波计算噪声分解系数的相关性，噪声本身的颜色必须知道。这方面的讨论已超出本书的范围，不再讲述。

（4）如果 s 是一个固定的零均值 ARMA 模型，则对每一个小波分解尺度 j，$C(j,k)(k \in z)$ 也是固定的、零均值的 ARMA 模型，它的特性取决于 j。

（5）如果 s 是一个噪声，并且它的相关函数 ρ 知道，我们就可以计算出系数序列 $C(j,k)$ 和 $C(j,k')$；如果其相关函数的谱 $\hat{\rho}$ 知道，则我们就可以计算出 $C(j,k)(k \in z)$ 的谱，以及不同尺度 j 和 j' 的交叉谱。

3. 一维小波分析对平稳信号消噪

例 3.6　某单位为了考察其用电情况，对其中三天的电网电压值进行监测，得到了一个电压采样序列，但由于在采样过程中，监测设备出现了一些故障，使得所采集的信号引入了噪声，现利用小波分析，将这种由于仪器故障引起的噪声进行消噪处理。

解：　该问题是一个消噪问题，利用小波消噪的基本原理，首先我们选择一个小波 db1，然后确定小波分解的层数 N（在这里，我们取 N 为 3）。从小波消噪处理的方法上说，一般有三种。

（1）强制消噪处理。该方法把小波分解结构中的高频系数全部变为 0，即把高频部分全部滤除掉，然后再对信号进行重构处理。这种方法比较简单，重构后的消噪信号也比较平滑，但容易丢失信号的有用成份。

（2）默认阈值消噪处理。该方法利用 ddencmp 函数产生信号的默认阈值，然后利用 wdencmp 函数进行消噪处理。

（3）给定软（或硬）阈值消噪处理。在实际的消噪处理过程中，阈值往往可以通过经验公式获得，而且这种阈值比默认阈值更具有可信度。在进行阈值量化处理中可用 wthresh 函数进行。

针对本问题，这里分别用上面三种消噪方法进行消噪处理，并对消噪的结果加以对比。

程序清单：

```
%采集的信号已经按照MATLAB格式存成MAT文件，文件名为leleccum.mat
load leleccum;  %将信号装入MATLAB工作环境
s=leleccum(1:3920); %取采样信号的前1～3920个采样点
ls=length(s); %计算采样序列长度
subplot(221); plot(s); %画出原始信号波形
title('原始信号');
[c,l]=wavedec(s,3,'db1');  %采用db1小波并对信号进行三层分解
ca3=appcoef(c,l,'db1',3); %提取小波分解的低频系数
cd3=detcoef(c,l,3); %提取第三层的高频系数
cd2=detcoef(c,l,2); %提取第二层的高频系数
cd1=detcoef(c,l,1); %提取第一层的高频系数
%下面对信号进行强制消噪处理
cdd3=zeros(1,length(cd3));
cdd2=zeros(1,length(cd2));
cdd1=zeros(1,length(cd1));
c1=[ca3,cdd3,cdd2,cdd1];
s1=waverec(c1,l,'db1'); %[c1,l]为新的分解结构
subplot(222); plot(s1); %画出对信号进行强制消噪的波形图
title('强制消噪波形图'); grid;
%下面利用默认阈值进行消噪处理
%用ddencmp函数获得信号的默认阈值，使用wdencmp命令函数来实现消噪过程
[thr,sorh,keepapp]=ddencmp('den','wv',s);
s2=wdencmp('gbl',c,l,'db1',3,thr,sorh,keepapp);
subplot(223);  plot(s2);
title('默认阈值消噪后的信号'); grid;
%下面利用给定的软阈值进行消噪
cd1soft=wthresh(cd1,'s',1.456);  %对第一层的高频系数，阈值取为
cd2soft=wthresh(cd2,'s',1.832);  %对第二层的高频系数，阈值取为
cd3soft=wthresh(cd3,'s',2.786);  %对第三层的高频系数，阈值取为

s3=waverec(c2,l,'dh1'); %[c2,l]为给定阈值量化后的分解结构
subplot(224); plot(s3);
title('给定软阈值消噪后的信号'); grid;
```

输出结果(如图3.8所示)。

从下面的图形我们可以看出，强制消噪处理后的信号较为光滑，但它有可能失去信号中的有用成份。而默认阈值消噪处理和给定阈值消噪处理则在实际应用中更实用一些。

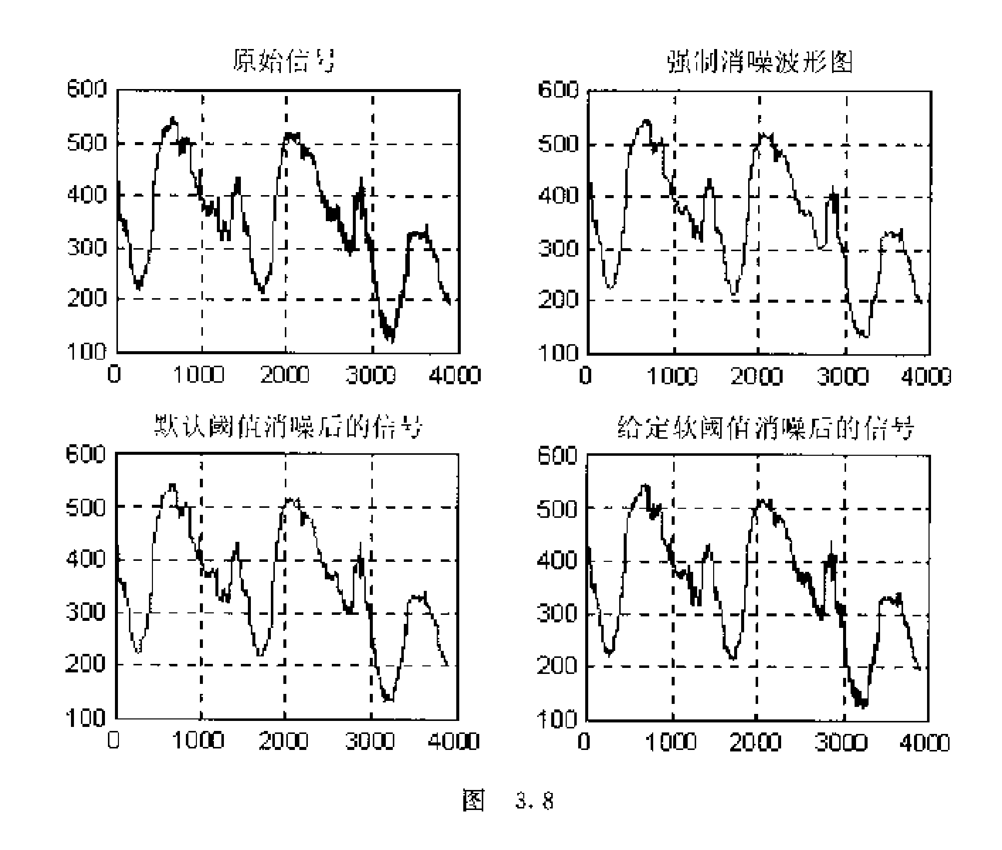


图 3.8

4. 一维小波分析对非平稳信号消噪

在实际的工程应用中，所分析的信号可能包含许多尖峰或突变部分，并且噪声也不是平稳的白噪声，对这种信号进行分析，首先需要作信号的预处理，将信号的噪声部分去除，提取有用信号。而这种信号的消噪，用传统的傅立叶变换分析，显得无能为力，因为傅立叶分析是将信号完全在频率域中进行分析，它不能给出信号在某个时间点上的变化情况，使得信号在时间轴上的任何一个突变，都会影响信号的整个谱图。而小波分析由于能同时在时频域中对信号进行分析(并且在频率域内分辨率高时，时间域内分辨率则低，在频率域内分辨率低时，时间域内分辨率则高，且有自动变焦的功能)，所以它能有效地区分信号中的突变部分和噪声，从而实现信号的消噪。下面将一个信号加入白噪声，然后用傅立叶分析方法和小波分析方法同时对加入白噪声的信号进行消噪处理，可以看出小波分析比传统的傅立叶分析更加优越。

例 3.7　产生一含噪声的矩形波信号，请分别用小波分析和傅里叶变换进行信号噪声消除。

解：该问题是一个非平稳信号噪声消除的问题，可按如下程序进行噪声消除。

程序清单：

```
snr=4; %设置信噪比
init=2055615866; %设置随机数初值
%产生原始信号 xref 和含白噪声信号 x
[xref,x]=wnoise(1,11,snr,init);
xref=xref(1:2000);
```

```
x=x(1:2000);
%用 sym8 小波进行三层分解并用 heursure 软阈值进行小波系数阈值量化
xd=wden(x,'heursure','s','one',3,'sym8'); %进行消噪处理
figure(1);
subplot(3,2,1); plot(xref);
title('原始信号');
subplot(3,2,2); plot(x);
title('含噪声信号');
%下面用傅里叶变换进行信号噪声消除
xxref=fft(xref);    %对原始信号进行傅立叶变换
xxref=abs(xxref);
xx=fft(x);    %对含噪信号进行傅立叶变换
absxx=abs(xx);
%下面画出傅里叶变换后的频谱图
figure(2);
subplot(3,2,1); plot(xxref);
title('原始信号的谱图');
subplot(3,2,2); plot(absxx);
title('含噪信号的谱图');
%进行低通滤波，滤波频率为 0～100 的相对频率
indd2=200:1800;
xx(indd2)=zeros(size(indd2));
xden=ifft(xx);    %进行傅立叶反变换
xden=real(xden);    %提取反变换后的实数部分
xden=abs(xden);
figure(3);
subplot(3,2,1); plot(xd);
title('小波消噪后的信号');
subplot(3,2,2); plot(xden);
title('用傅立叶分析消噪的效果')
```

输出结果(如图 3.9 所示)。

从下图的比较中，可以看出，用小波进行信号的消噪可以很好地保存有用信号中的尖峰和突变部分。而用傅立叶分析进行滤波时，由于信号集中在低频部分，噪声分布在高频部分，所以，可用低通滤波器进行滤波，但是，它不能将有用信号的高频部分和由噪声引起的高频干扰加以有效地区分。若低通滤波器太窄，则在滤波后，信号中仍存在大量的噪声，若低通滤波器太宽，则将一部分有用信号当作噪声而滤掉了。因此，小波分析对非平稳信号消噪有着傅里叶分析不可比拟的优点。

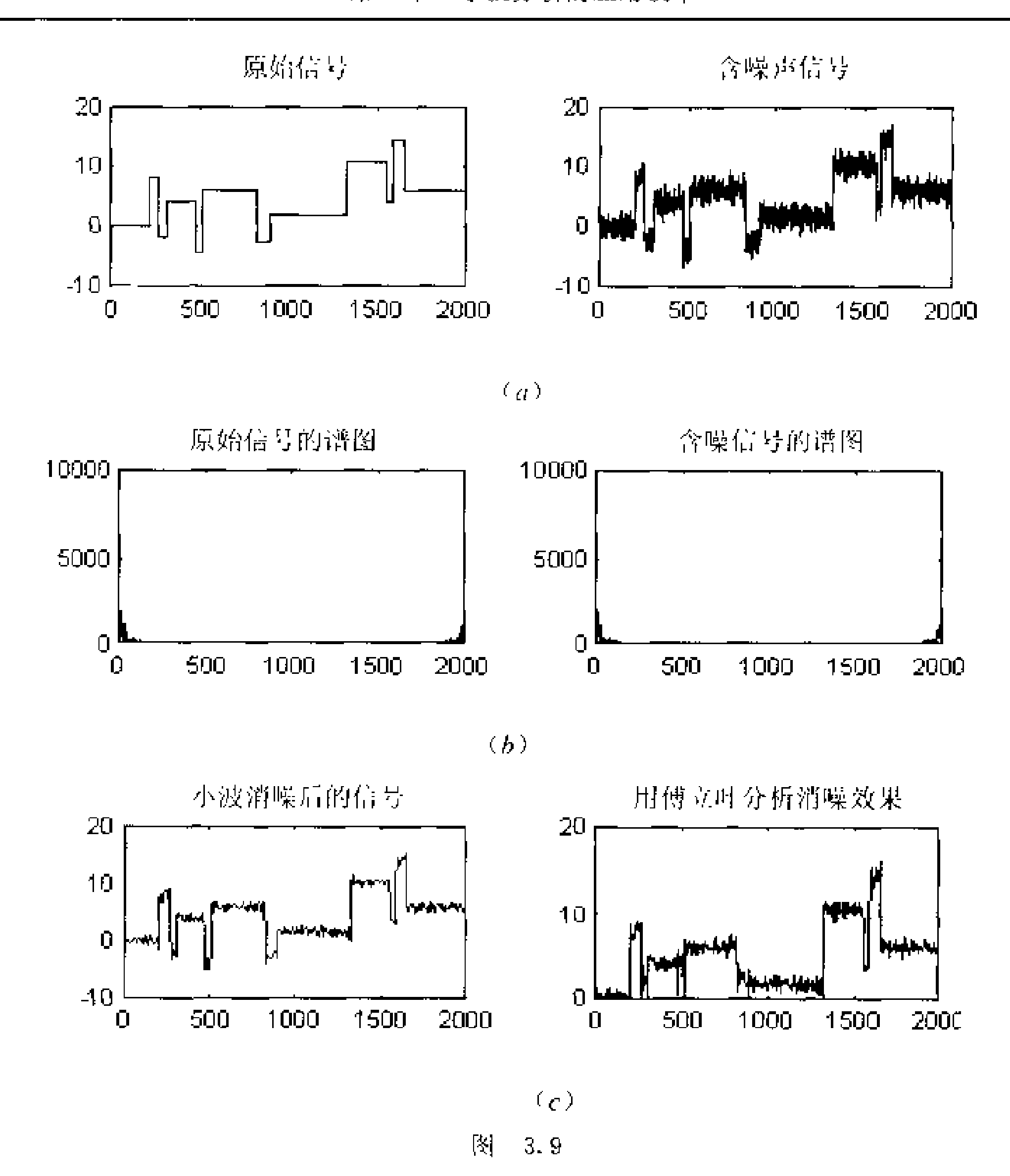


图　3.9

5. 小波消噪中阈值的选取规则

在用一维小波进行信号的消噪和压缩过程中，都要用阈值进行小波分解系数的量化处理，在这两种信号处理中，最重要的环节就是如何选取阈值和如何进行阈值的量化。所以下面对阈值的选取原则进行说明。

根据基本的噪声模型，可以在 thselect 函数中使用四种规则来选取阈值，每一种规则的选择由该函数中所对应的输入参数 tptr 决定。该函数返回的是所求阈值的值。tptr 有四种规则，具体见表 3－7。

阈值选择规则是基于基本模型 $y=f(t)+e$，其中 e 是白噪声 $N(0,1)$，对于方差未知的噪声或非白噪声可以重新调节输出阈值 THR（参见 wden 中的 SCAL 参数）。

可以选择以下几种阈值：

• tptr＝'rigrsure'是一种基于史坦的无偏似然估计（二次方程）原理的自适应阈值选择。对一个给定的阈值 t，得到它的似然估计，再将非似然 t 最小化，就得到了所选的阈值，它是一种软件阈值估计器。

表 3-7 参数 tptr 的四种选项

tptr 的选项	阈值选择规则
'rigrsure'	采用史坦(Stein)的无偏似然估计(Unbiased Risk Estimate)原理(SURE)进行阈值选择
'sqtwolog'	固定的阈值形式，它等于 sqrt(2 * log(length(s)))
'heursure'	启发式阈值选择
'minimaxi'	用极大极小原理选择的阈值

• tptr='sqtwolog'采用的是固定的阈值形式，产生的阈值大小是 sqrt (2 * log(length(X)))。

• tptr='heursure'是前两种阈值的综合，是最优预测变量阈值选择。如果信噪比很小，SURE 估计有很大的噪声。如果有这种情况，就采用这种固定的阈值。

• tptr='minimaxi'采用的也是一种固定的阈值，它产生一个最小均方误差的极值，而不是无误差。在统计学上，这种极值原理用于设计估计器。因为被消噪的信号可以看作与未知回归函数的估计式相似，这种极值估计器可以在一个给定的函数集中实现最大均方误差最小化。

如果信号 y 表示的是一个高斯白噪声信号 $N(0,1)$，那么，各个阈值选择规则下的 thselect 命令函数是如何选取阈值的。

例 3.8 下面给出一个白噪声信号(信号采样点的个数为 1000)，请计算四种阈值选择的阈值大小。

解：四种阈值的计算可按如下的程序进行。

```
%程序清单：
y=randn(1,1000);
thr1=thselect(y,'rigrsure')
thr2=thselect(y,'sqtwolog')
thr3=thselect(y,'heursure')
thr4=thselect(y,'minimaxi')
    输出结果：
thr1 =
    2.7316
thr2 =
    3.7169
thr3 =
    3.7169
thr4 =
    2.2163
```

由于 y 是一个标准的高斯白噪声，我们希望每一种方法都能将系数中的大部分数值去掉。对于 Stein 的无偏似然估计(SURE)和极大极小原理(minimaxi)的阈值选择规则，只大

约保存了 3%的系数，而对于另外两种阈值选取规则，所有的系数都被变成了零。

我们知道，当对噪声 e 进行小波分解时，它同样会产生高频系数，所以，一个信号的高频系数向量是有用信号 f 和噪声信号 e 的高频系数的叠加。由于 minimaxi 和 SURE 阈值选取规则比较保守(它只将部分系数置 0)，当信号 f 的高频信息有很少一部分在噪声范围内时，这两种阈值非常有用，可以将弱小的信号提取出来；另外两种阈值选取原则，在去除噪声时，显得更为有效，但它有可能把有用的高频特征信号当做噪声信号消除。

3.1.5　识别在含噪信号中有用信号的发展趋势

例 3.9　一个有用的斜坡信号由于被有色噪声(即能量有限噪声)所污染，(信号的文件名为：conislop.mat)使有用信号的斜线趋势在时域中看不出来，请利用小波分析出这个信号的整体发展趋势。

解：在这个例子中，我们用一个 db3 小波的六层分析，使斜坡信号的斜坡发展趋势随着层次的升高，表现得越来越清楚。下面是实现这一过程的程序清单。

程序清单：

```
load cnoislop; %装入要分析的信号
s=cnoislop; ls=length(s);
[c,l]=wavedec(s,6,'db3'); %用 db3 小波分解信号到第六层
subplot(7,1,1); plot(s);
title('原始信号及各层重构信号');
Ylabel('s');
%对分解结构[c,l]中的各低频部分进行重构，并显示结果
for i=1:6
    decmp=wrcoef('a',c,l,'db3',7-i);
    subplot(7,1,i+1);
    plot(decmp);
    Ylabel(['a',num2str(7-i)]);
end
```

输出结果(如图 3.10 所示)。

信号中低频部分代表着信号的发展趋势，在小波分析中，则对应着最大的尺度小波变换的低频系数。随着尺度的增加，时间分辨率的降低，对信号的这种发展趋势会表现得更明显。另外，我们还可在频率中理解它，尺度分解中的低频部分随着层次的增加，它含有的高频信息会随之减少。当分解到下一个层次时，就有更高频率的信息被去除，则剩下的就是信号的发展趋势。

从这里可以看出，小波分析在展示信号的发展趋势中很有用，另外，这种分析还有一个目的就是将藏在噪声中的信号显示出来。需要强调的是，信号的发展趋势代表的是信号中最低的频率部分，如果信号本身包含有很陡的变化，那么在多尺度小波变换的低频部分中，显示的信号则越来越不像原始信号，因为它将信号本身的陡峭变化当作高频滤掉了。

例 3.10　对某型号导弹上的加速度表进行测试时，主要评定加速度表的极性及其灵敏度。加速度表的正常与故障往往表现在它的低频信号(细节信号)的发展趋势不同。现用

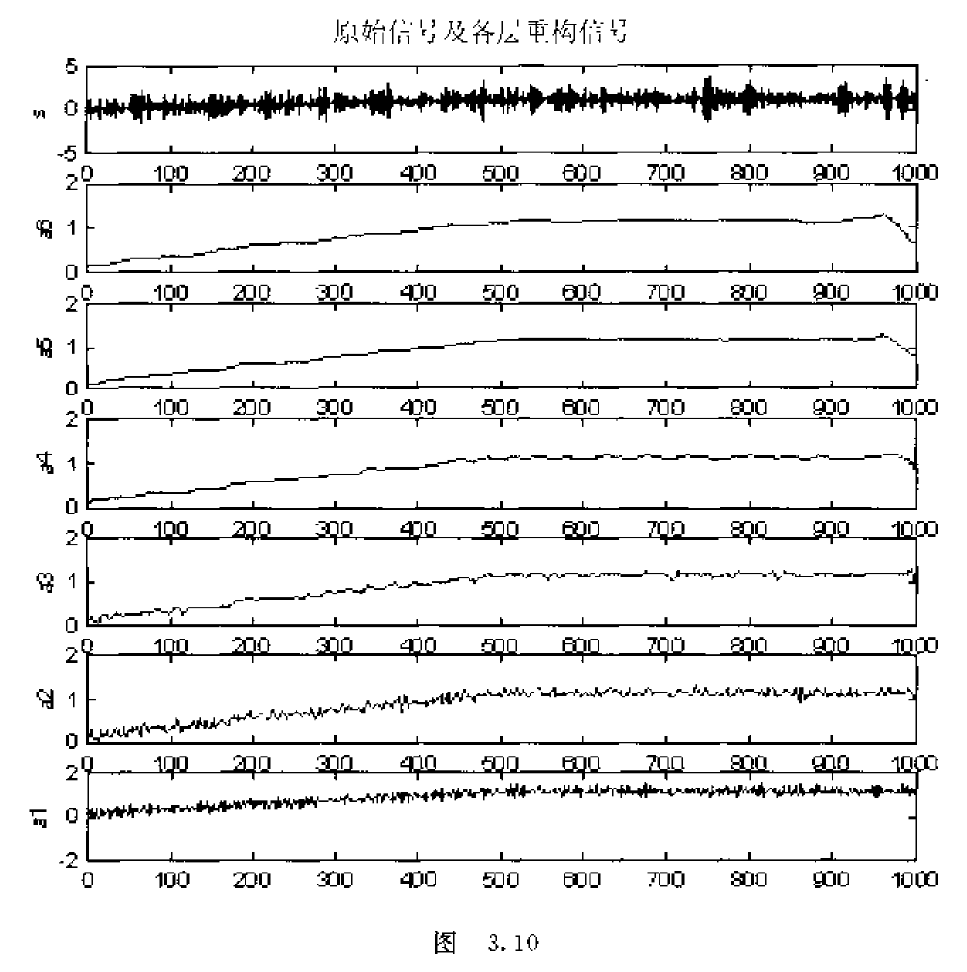


图　3.10

A/D 采样板分别采样得到加速度表在正常和故障两种状态时的两组数据，但由于某种原因使得测试数据中混入了噪声干扰，因而正常数据和故障数据的波形不易区分，试用小波分析对加速度表正常与故障时信号的发展趋势加以区分。

解：用 A/D 采样板采样得到正常与故障的两组数据分别为：a_152ms.dat 和 a_162ms.dat，现对这两组信号分别进行小波变换，然后提取信号的低频成份（细节部分），即可以判定信号的发展趋势。该问题的处理可编程如下：

```
load d:\zjb\missiledata\a_152ms.dat;
load d:\zjb\missiledata\a_162ms.dat;
s1=a_152ms(1:10:1000);
s2=a_162ms(1:10:1000);
subplot(321); plot(s1);
title('正常情况及其发展趋势');
subplot(322); plot(s2);
title('故障情况及其发展趋势');
[c,l]=wavedec(s1,6,'db3');      %用 db5 小波分解信号到第六层
a6=wrcoef('a',c,l,'db3',6);
subplot(323); plot(a6); Ylabel('a6');
[c,l]=wavedec(s2,6,'db3');      %用 db5 小波分解信号到第六层
```

```
a6=wrcoef('a',c,l,'db3',6);
subplot(324); plot(a6); Ylabel('a6');
```

输出结果(如图 3.11 所示):

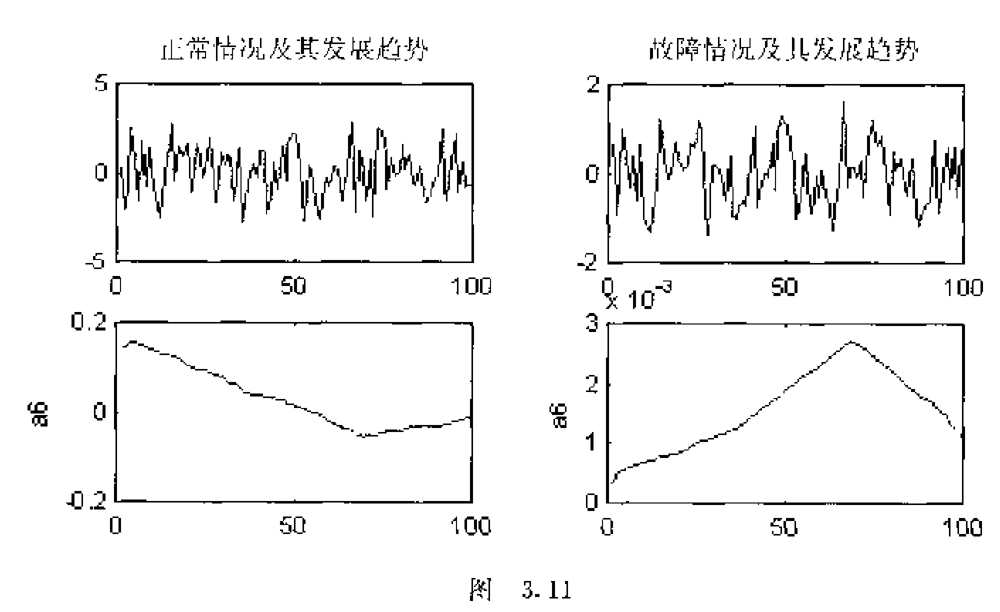


图 3.11

从第六层的细节信号，我们可以明显看出正常情况和故障情况的信号发展趋势是完全不一样的，但如果从原始图形上观察，则很难判断信号的发展趋势。

3.1.6 提取信号中某一频率区间的信号

在傅立叶分析中，如果一个信号是由几个不同频率的正弦信号所组成，则 FFT 可以很有效地分辨这些不同频率的正弦信号。在这个例子中，我们用小波分析来实现这一功能。所分析的信号是由三个不同频率的正弦信号所组成。

例 3.11 给定一信号，请利用小波分析把信号的各个频率成份分开。

程序清单：

```
x=0:0.05:6*pi;
s=sin(x)+sin(10*x)+sin(100*x);        %产生一个正弦叠加信号
figure(1);
subplot(6,2,1); plot(s);
title('原始信号与各层低频部分');
Ylabel('s');
subplot(6,2,2); plot(s);
title('原始信号与各层高频部分');
Ylabel('s');
[c,l]=wavedec(s,5,'db3');       %用 db3 小波分解信号到第五层
%对分解结构[c,l]中各低频部分进行重构,并显示结果
for i=1:5
    decmp=wrcoef('a',c,l,'db3',6-i);
    subplot(6,2,2*i+1);
```

```
    plot(decmp);
    Ylabel(['a',num2str(6 -i)]);
end
%对分解结构[c,l]中各高频部分进行重构,并显示结果
for i=1:5
    decmp=wrcoef('d',c,l,'db3',6-i);
    subplot(6,2,2*i+2);
    plot(decmp);
    Ylabel(['d',num2str(6-i)]);
end
%画出 d1 的放大波形图
figure(2);
d1=wrcoef('d',c,l,'db3',1);
subplot(411); plot(d1(1:100));
```

输出结果(如图 3.12 所示):

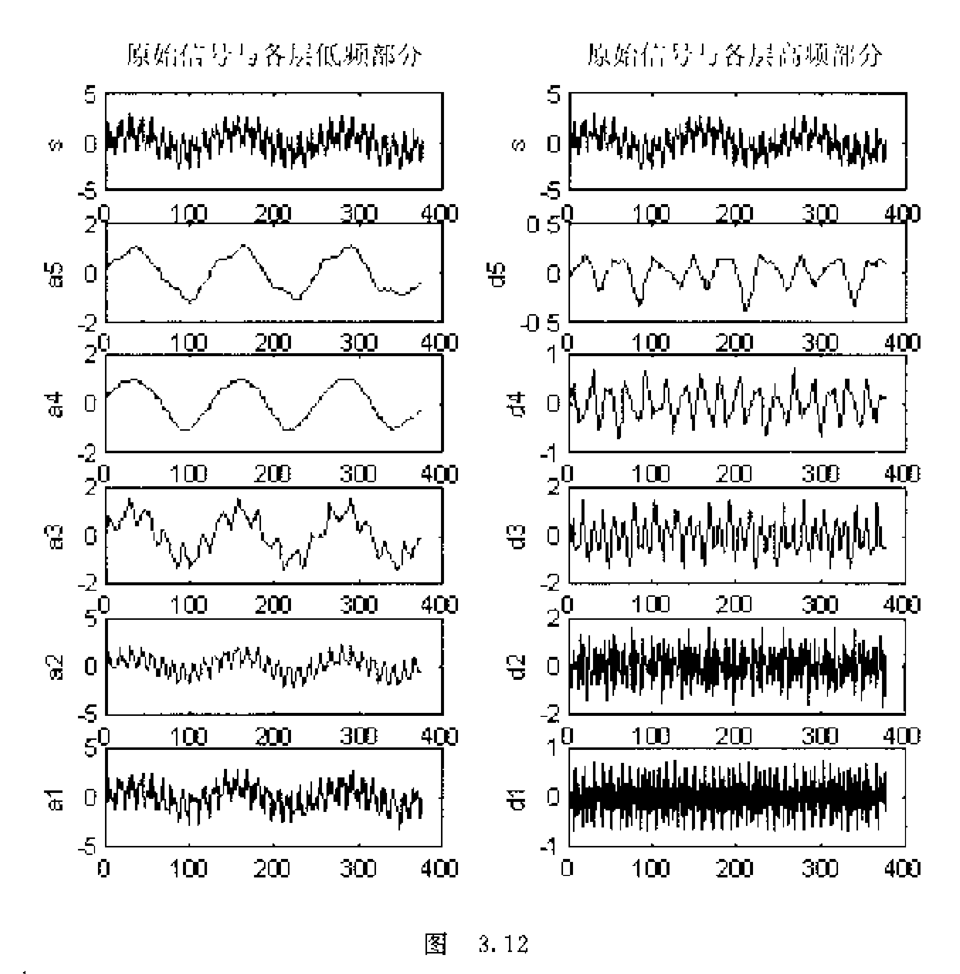


图 3.12

在显示的分解结果中，可以看到，低频的第四层将正弦信号中的最低频率组成清晰地分离出来了。

在小波分解中，若将信号中的最高频率成分看作是 1，则各层小波分解便是带通或低

通滤波器，且各层所占的具体频带为：

a1：0 — 0.5　　d1：0.5 — 1

a2：0 — 0.25　　d2：0.25 — 0.5

a3：0 — 0.125　　d3：0.125 — 0.25

a4：0 — 0.0625　　d4：0.0625 — 0.125

a5：0 — 0.03125　　d5：0.03125 — 0.0625

该信号是由三个不同频率的正弦信号所组成，其频率在小波分解下的相对频率分别为：高频正弦(1)、中频正弦(0.1)、低频正弦(0.01)。由信号频率的组成部分和小波分解下各层频率的分布可知，高频正弦定位于 d1 层，实际上也是如此。左上图显示的是放大后的 d1 层信号图。从该图可以看出，每一个信号包络中，有 10 个正弦振荡，它的相对周期的估计值为 200(在小波分析中的频率为 1)。而 d4 层则包含中频层的正弦信号。我们注意到在 a3 层和 a4 层中间有不连续的点出现，因为中频正弦信号是由这两层共同表达的，d4 层的信息有一部分被 d3 层减去了。我们需要用 a1 到 a3 层的信息来估计中频层的正弦信号，将 a1 放大后可以看出中频层的正弦信号，其相对周期大约是 20(在小波分析中的频率为 0.1)。最后，只剩下低频正弦信号没有提取出来，在 a4 层中对其进行估计，可以看出其相对周期大约是 2，一个周期占 200 个点(在小波分析下的频率为 0.01)。其实，低频正弦信号在 a5 层也能看出来，但是，这就必须将信号进行小波的进一步分解。若进一步分解，可以看到，该信号在 a4 层消失了，而出现在 d8 层。

总之，我们能够用小波分析将合成信号中的单纯正弦信号的频率提取出来。因为在小波分解下，不同的尺度具有不同的时间和频率分辨率，因而小波分解能将信号的不同频率成份分开。

3.1.7　进行信号某些频率区间的抑制或衰减

小波分析进行信号抑制是小波分析实际应用中的一个重要的方面。一般说来，一个信号可以用一个次数为 k 的多项式来逼近表示，所以，小波对信号的抑制可以归结为对一个多项式幅值的抑制，小波抑制一个多项式幅值的能力取决于小波本身的一个重要的数学特征，该特征称为**过零点**。在这里，可以将过零点理解为一种平均值的外延。如果一个小波的平均值是 0，则它至少有一个过零点。其精确的数学描述为：

如果 $\mathrm{mean}(x^k\Psi(x))=0,(k=0,\cdots,n;\ \Psi(x)$是小波函数；mean 表示平均值)，则该小波函数有 $n+1$ 个过零点，它可以抑制一个次数为 n 的多项式。

如果 Ψ 是一个有至少 $n+1$ 个过零点的小波函数(即对 $j=0,\cdots,k$，有$\int_R x^j\Psi(x)\mathrm{d}x=0$) 且信号 s 是一个 k 次的多项式，则系数 $C(a,b)=0$ 对所有的 a 和 b 都成立，那么这种小波函数可以自动抑制多项式信号，这时，信号 s 的次数可以随时间变化，只要保证它的次数小于 k 即可。

如果信号 s 是一个只在$[\alpha,\beta]$区间段是次数为 k 的多项式，则只要函数$\frac{1}{\sqrt{a}}\Psi\left(\frac{x-b}{a}\right)$在区间$[\alpha,\beta]$内成立，系数 $C(a,b)=0$ 也成立，那么小波分析就可以进行信号的抑制，但在信号段的边界部分有衰减作用。所以，我们可以假设在区间$[\alpha,\beta]$中，信号 $s(x)$可以展开成 $s(x)=s(0)+xs(0)+x^2s^{(2)}(0)+\cdots+x^ks^{(k)}(0)+g(x)$，则 s 和 g 有相同的小波系数。这里可

以用相位的观点来理解：小波只抑制信号 s 中的多项式部分，而对信号 s 中的"不规则"信号 g 不起作用。小波分析通过对规则信号的抑制来达到对不规则信号进行分析的目的。

抑制信号中某些成份的另一种方法是将小波分解系数中的某些系数 $C(a,b)$ 强制性地等于 0。对于一个系数组成的集合 E，我们有 $\forall\ (a,b) \in E$，令 $c(a,b)=0$ ，然后，再将修改后的小波分解系数进行小波重构，则可以对该信号的某些部分进行抑制。这种方法，通过下面的例子来具体说明。

例 3.12 给定一含噪声的正弦信号（文件名为：sumsin.mat），请利用小波分析对信号的高频部分进行信号抑制。

解：这里以三个叠加的正弦信号来进行分析说明。首先进行小波分解，通过将第三层和第四层的高频系数全部置零来实现对时间段[400,600]内的高频部分进行高频抑制；同时，又将信号在 $t=500$ 的时间点进行了放大，方法是令这一点对应的第二层高频系数等于 4。这种方法可以达到良好的信号抑制效果。

```
%程序清单：
load sumsin;
s=sumsin(1:1000);
w='coif3'; maxlev=4; %设置所用的小波和分解的层数
[c,l]=wavedec(s,maxlev,w);
%将小波分解后的第三层、第四层的高频系数全部置 0
newc=wthcoef('d',c,l,[3,4]);
%将第一层的高频系数在原始信号时间段[400,600]区间内的系数置 0，并且将其
%它系数进行衰减，求出第一层系数的初始点和最后点的索引
k=maxlev+1;
first=sum(l(1:k-1))+1; last=first+l(k)-1;
indd1=first:last;
newc(indd1)=c(indd1)/3; %将系数除以 3，进行信号的衰减
%在时间段[400:600]内求出第一层系数的索引，注意到原始信号的时间点 t 对应
%第 k 层的时间点 t/(2^k)，在这里，k=1
indd1=first+400/2:first+600/2;
newc(indd1)=zeros(size(indd1)); %将这些系数值置 0
%将与原始信号的时间点 t=500 大致对应的第二层高频系数的值变成 4
k=maxlev; first=sum(l(1:k-1))+1;
newc(first+500/2^2)=4;
%将修改后的小波分解系数结构的值进行合成
synth=waverec(newc,l,w);
subplot(2,2,1); plot(s);
title('原始信号');
subplot(2,2,2); plot(c);
title('小波分解后的系数');
subplot(2,2,3); plot(synth);
```

```
title('用小波抑制后的信号');
subplot(2,2,4); plot(newc);
title('修改后的小波分解系数');
```

输出结果(如图 3.13 所示)：

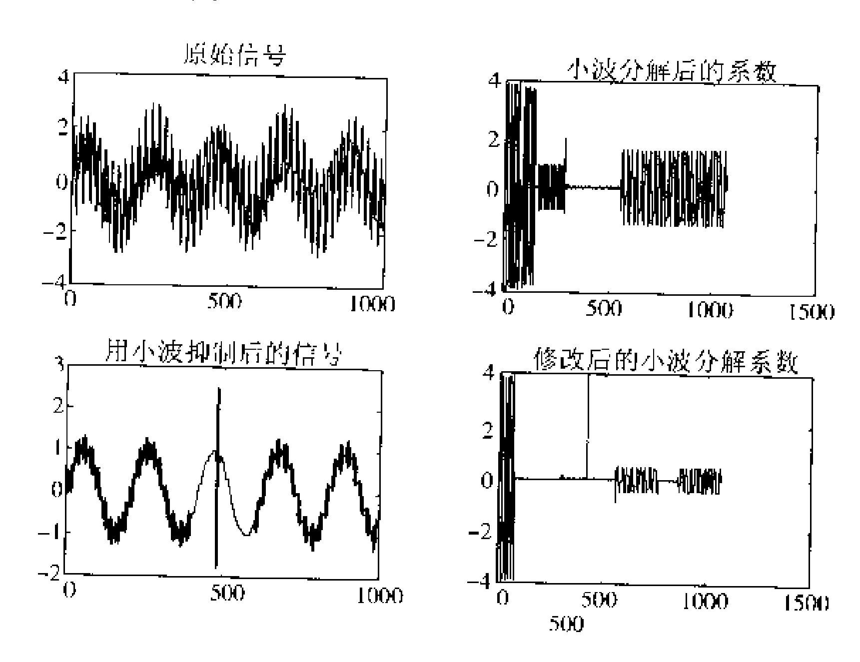


图　3.13

对一个多项式用小波进行分解时，可以导致信号分解的高频部分为 0，这是因为小波的过零点的个数超过了多项式的次数。下面用一个二次多项式加上一些白噪声来说明这一问题。

例 3.13　给定一信号，请利用小波分析对信号某些频率故意进行衰减。

解：该问题可进行如下处理。

程序清单：

```
load noispol; %装入要分析的信号
s=noispol; ls=length(s);
[c,l]=wavedec(s,4,'db3'); %用 db3 小波分解信号到第四层
figure(1);
subplot(5,1,1); plot(s);
title('原始信号及用 db3 小波分解其各层高频信号重构图');
Ylabel('s');
%对分解结构[c,l]中各高频部分进行重构,并显示结果
for i=1:4
    decmp=wrcoef('d',c,l,'db3',5-i);
    subplot(5,1,i+1);
    plot(decmp);
    Ylabel(['d',num2str(5-i)]);
end
```

```
%下面用 db1 小波进行对照分析
[c,l]=wavedec(s,4,'db1'); %用 db1 小波分解信号到第四层
figure(2);
subplot(5,1,1); plot(s);
title('原始信号及用 db1 小波分解其各层高频信号重构图');
Ylabel('s');
%对分解结构[c,l]中各高频部分进行重构,并显示结果
for i=1:4
    decmp=wrcoef('d',c,l,'db1',5-i);
    subplot(5,1,i+1);
    plot(decmp);
    Ylabel(['d',num2str(5-i)]);
end
```

输出结果(如图 3.14 所示)。

从 db3 小波和 db1 小波的对照分析可以看出，由于 db3 小波有三个过零点，因而尺度 2(或更大)的高频系数几乎为 0，而 db1 小波只有一个过零点，其分解的高频系数表现出明显不同的特性。

3.1.8 检测信号的自相似性

例 3.14 给定一信号(信号的文件名为：wonkoch.mat)，请利用小波分析检测信号的自相似性。

解：用小波检测信号的自相似性，即检测信号的分形特征。这里用到的信号是一个 Koch 信号，即一个经过反复迭代得到的合成信号。在这里我们用一维连续小波对该迭代信号进行分解。下面是分解的程序清单。

```
load vonkoch;    s=vonkoch;  %装入信号
subplot(2,1,1); plot(s);
title('原始信号');
%进行一维连续小波变换，把返回系数存到矩阵 w 中
subplot(2,1,2);
w=cwt(vonkoch,[2:2:128],'coif3','plot');    %分解尺度为 2~128，以 2 递增
Xlabel('时间'); Ylabel('变换尺度');
title('小波分解的系数图');
%输出结果(如图 3.15 所示)。
```

在分解后显示的图形中，会发现小波系数的图形在许多尺度看上去很相似。从直观上看，小波分解可通过计算信号和小波之间的“自相似指数”(Resemblance Index)来得到。如果“自相似指数”很大，则信号的自相似程度就很大，反之则很小。这种指数就是小波系数。如果一个信号在一个不同的尺度上都与它自己相似，则“自相似指数”或小波系数也在不同的尺度上很相似。在小波系数图中，垂直轴线上显示的线条就是由于信号的自相似性而产生的。

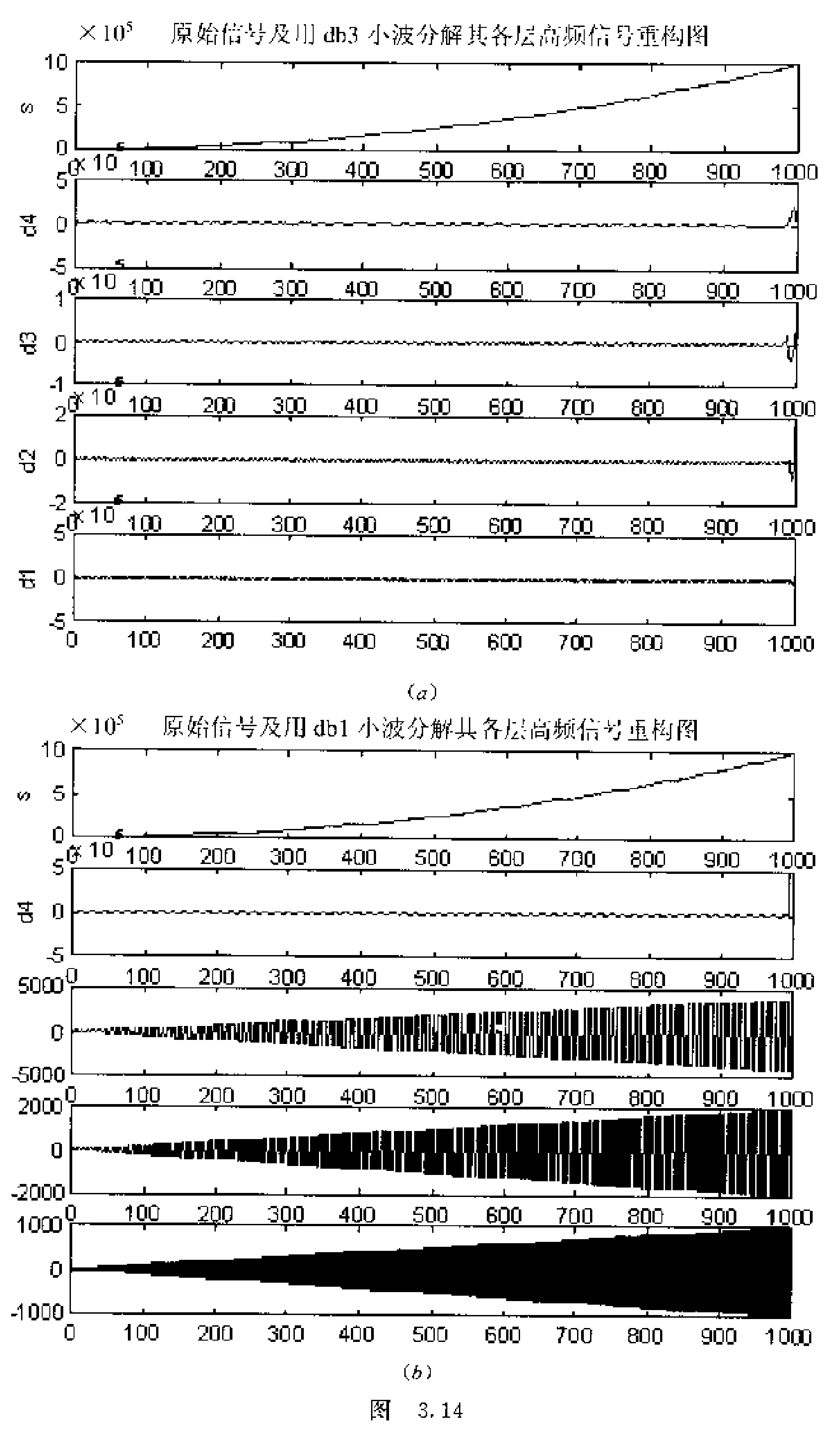


图 3.14

现在，许多人正在做这项工作，其结果表明，采用小波分解，可以很好地研究信号或图像的分形特征。当分形的特征开始时随着时间的发展而变化，然后又不变时，则这种信号被称为多分形(multifractal)。小波分析工具非常适合于分形的实际研究和分形的生成。

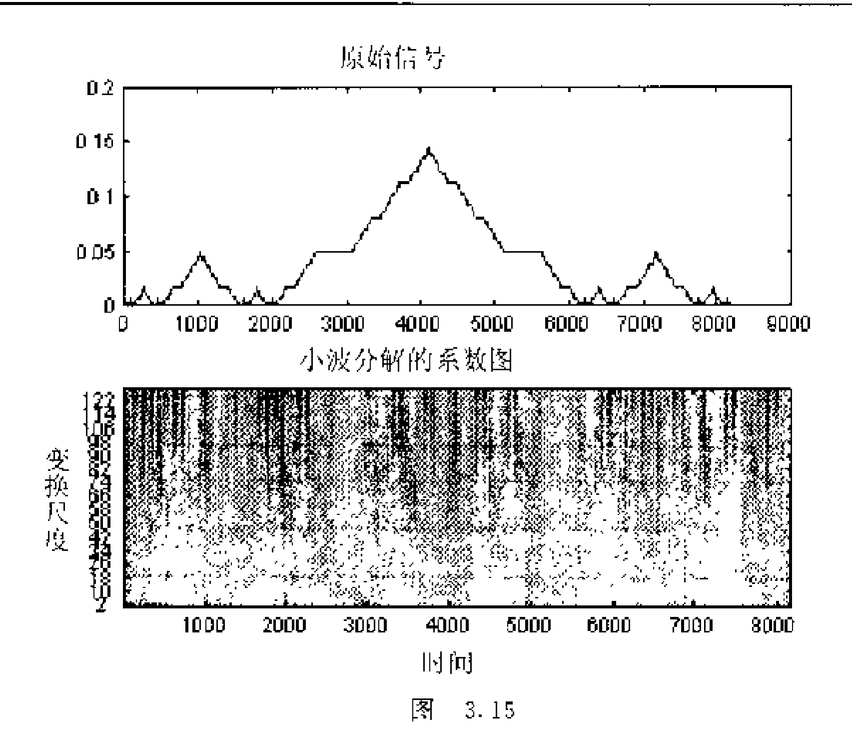


图　3.15

3.1.9　一维小波分析进行信号压缩

· 一维小波分析进行信号压缩的原理

用一维小波之所以能够对信号进行压缩，是因为一个规则信号的组成可以用下面的几个部分来精确地逼近，即一个数据量很小的低频系数(该低频系数需要选在一个合适的分解层上)和几个高频层的系数。

· 一维小波分析进行信号压缩的一般步骤

与信号消噪类似，信号的压缩也分为以下几个步骤：

(1) 信号的小波分解。

(2) 将高频系数进行阈值量化处理。对从 1 到 N 的每一层高频系数，都可以选择不同的阈值，并且用硬阈值进行系数的量化。

(3) 对量化后的系数进行小波重构。

信号的压缩与消噪的不同之处在第二步。一般说来，有两种比较有效的信号压缩方法，第一种方法是对信号进行小波尺度的扩展，并且保留绝对值最大的系数。在这种情况下，可以用全局阈值来压缩信号，来实现信号的压缩或相对均方误差规范的信号恢复，这时只需用一个输入参数；第二种方法是根据每一层分解后的效果，来决定该层的阈值，而且每层的阈值是不同的。

例 3.15　对于某一给定的信号(信号的文件名为 leleccum.mat)，利用小波分析对信号进行压缩处理。

解：处理过程可以编程如下：

```
load leleccum；     %将信号装入 MATLAB 工作环境
%设置变量名 s 和 ls，在原始信号中，只取 2600～31000 个点，其余点不予考虑
s=leleccum(2600：3100)；   ls=length(s)；
[c,l]=wavedec(s,3,'db3')；    %用 db3 小波分解信号到第三层
```

```
%选用全局阈值进行信号压缩。
thr=35;
[xd,cxd,lxd,perf0,perfl1]=wdencmp('gbl',c,l,'db3',3,thr,'h',1);
subplot(2,1,1); plot(s);
title('原始信号 s');
subplot(2,1,2); plot(xd);
title('压缩后的信号 xd');
```

输出结果(如图 3.16 所示):

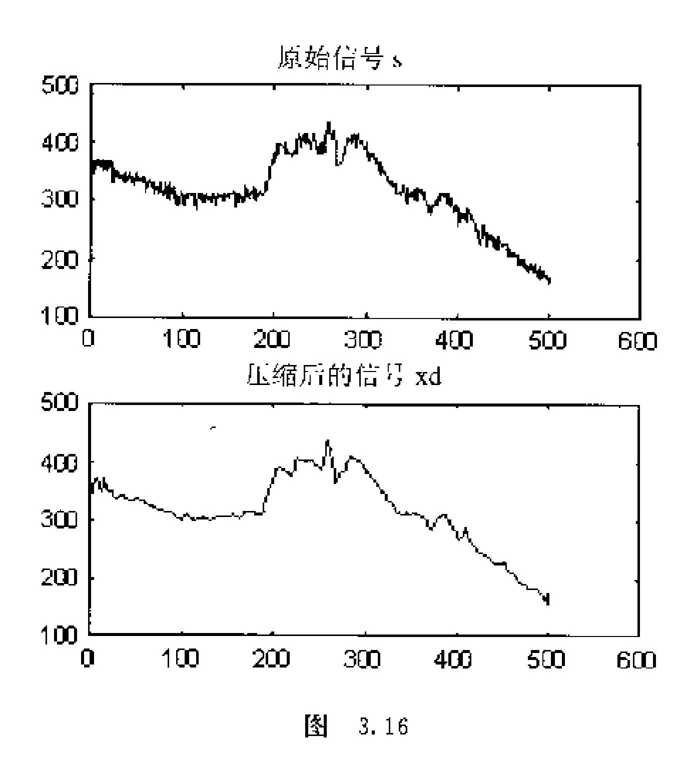


图　3.16

3.2　二维小波分析的应用

在本节，我们主要讲解二维小波分析在实际工程中的应用情况。主要以二维小波变换在图像处理中的应用为例加以说明。首先，我们将用于二维小波图像分析的主要函数作一个简要介绍，每个函数的具体用法，请参阅第 2 章的有关内容。用于图像分析的函数主要有：

(1) 小波分解函数

表 3-8　二维小波分解函数

函数名	功　能
dwt2	单尺度(层)二维小波分解
dwtper2	单尺度二维离散小波变换(周期性)
wavedec2	多尺度(层)二维小波分解

(2) 小波重建函数

表 3-9　二维小波重建函数

函数名	功　　能
idwt2	单尺度二维小波重构
idwtper2	单尺度二维离散逆小波分析(周期性)
waverec2	多尺度二维小波重构
upwlev2	二维小波分解的单尺度重构
wrcoef2	二维小波分解系数单支重构
upcoef2	二维小波分解的直接重构

(3) 分解结构应用函数

表 3-10　二维小波分解结构应用函数

函数名	功　　能
detcoef2	提取二维小波分解高频系数
appcoef2	提取二维小波分解低频系数

(4) 消噪和压缩函数

表 3-11　二维小波消噪和压缩函数

函数名	功　　能
wthresh	进行软阈值或硬阈值处理
wthcoef2	二维信号的小波系数阈值处理
ddencmp	获取在消噪或压缩过程中的默认值阈值(软或硬)、熵标准
wdencmp	用小波进行信号的消噪和压缩(也可以用于一维的情况)

3.2.1　二维小波分析用于图像压缩

1. 问题提出

对于图像来说，如果需要进行快速或实时传输以及大量存储，就需要对图像数据进行压缩。在同等的通信容量下，如果图像数据压缩后再传输，就可以传输更多的图像信息，也就可以增加通信的能力。例如，用普通的电话线传输图像信息。图像压缩研究的就是寻找高压缩比的方法且压缩后的图像要有合适的信噪比，在压缩传输后还要恢复原信号，并且在压缩、传输、恢复的过程中，还要求图像的失真度小，便于图像的分类、识别等。

2. 解决途径

图像数据往往存在各种信息的冗余，如空间冗余、信息熵冗余、视觉冗余和结构冗余

等。所谓压缩就是去掉各种冗余，保留对我们有用的信息。图像压缩的过程常称为编码。图像恢复的过程常称为解码。

根据解码后的数据与原始数据是否完全一致来分类，图像压缩方法一般划分为无失真编码与有失真编码两大类。无失真编码方法要求在解码后得到的图像与原始图像严格相同，如改进的 Huffman 编码。

有失真编码方法的还原图像较之原始图像存在一定的误差，但视觉效果一般是可以接受的。根据有失真编码的原理进行分类，可以有预测编码、变换编码、量化编码、信息熵编码、分频带编码、结构编码及基于知识的编码等。

预测编码是一种针对统计冗余进行压缩的方法。对于统计冗余来说，它反映图像内相邻两像素之间的相关性比较强，因而一个像素可以由与它相邻的并且已被编码的像素来进行预测估计，当然，预测是根据某一模型进行的。从理论上讲，只要模型选取得足够好，则只需存储或传输起始像素与模型参数就可以代替一幅图像。

量化与向量量化编码的本质也是针对统计冗余进行压缩，不过从表现形式上看，好像是无关的。

信息熵编码是根据信息熵原理，让出现概率大的用短的码字表达，反之用长的码字表达，最常见的如 Huffman 编码、游程编码以及算术编码等。

分频带编码是将图像数据变换到频域后，按频率分频带，然后用不同的量化器进行量化，从而达到最优的组合。或者是分步渐进编码，即开始对某一频带的信号进行解码，然后扩展到所有频带。随着解码数据的增加，解码图像也逐渐清晰起来。此方法对于远距离图像模糊查询与检索比较有效。

结构编码也称第二代编码。编码时首先将图像中的边界、轮廓、纹理等结构特征求出来，然后保存这些参数信息。解码时，根据结构和参数信息进行反合成，从而恢复图像。

基于知识的编码主要用于可规则描述的图像，例如人脸，利用人们对人脸的知识形成一个规则库，据此将人脸的变换用一些参数进行描述，从而用参数与模型就可以实现人脸的图像编码与解码。

变换编码也是一种针对统计冗余进行压缩的方法。所谓变换编码是将时域图像(空间)变换到系数域(频域)上进行处理的方法。因为由时域映射到频域总是通过某种变换进行的，所以称为变换编码方法。在空间上具有强相关的信号，反映在频域上是在某些特定的区域中能量集中在一起，或者是系数矩阵的分布具有某种规律，这就可以利用这些规律分配频域上的量化比特数，从而达到压缩的目的。常用的变换有 KL 变换、DCT、DST 变换；DFT 变换；Haar 变换；Walsh－Hadamard 变换以及用途广泛的小波变换等；变换编码有两个般明显的特点，一是可以得到高的压缩比，二是比预测等其它方法的计算复杂性高。在变换后，由于在频域上信息是按照频谱的能量与频率分布排列的，只要对频域平面量化器进行合理的(非均匀)比特分配，高能量区给以高的比特数，低能量区给以低的比特数，就可以得到高的压缩能力。

小波分析用于信号与图像压缩是小波分析应用的一个重要方面。它的特点是压缩比高，压缩速度快，压缩后能保持信号与图像的特征基本不变，且在传递过程中可以抗干扰。

基于小波分析的图像压缩方法很多，比较成功的有小波包最好基方法、小波域纹理模型方法、小波变换零树压缩、小波变换向量量化压缩等。

例 3.16 给出一个图像(即一个二维信号，文件名为 wbarb.mat)，请利用二维小波分析对图像进行压缩。

解：该问题是一个利用小波分析进行图像压缩的实例。一个图像作小波分解后，可得到一系列不同分辨率的子图像，不同分辨率的子图像对应的频率是不相同的。高分辨率(即高频)子图像上大部分点的数值都接近于 0，越是高频这种现象越明显。对一个图像来说，表现一个图像最主要的部分是低频部分，所以一个最简单的压缩方法是利用小波分解，去掉图像的高频部分而只保留低频部分。图像压缩可按如下程序进行处理。

程序清单：

```
clear %清除 MATLAB 工作环境中现有的变量
load wbarb; %装入图像
%显示图像
subplot(221); image(X); colormap(map)
title('原始图像');
axis square
disp('压缩前图像 X 的大小:');
whos('X')
%==================================
% 对图像用 bior3.7 小波进行 2 层小波分解
[c,s]=wavedec2(X,2,'bior3.7');
%提取小波分解结构中第 1 层的低频系数和高频系数
ca1=appcoef2(c,s,'bior3.7',1);
ch1=detcoef2('h',c,s,1); %水平方向
cv1=detcoef2('v',c,s,1); %垂直方向
cd1=detcoef2('d',c,s,1); %斜线方向
%分别对各频率成份进行重构
a1=wrcoef2('a',c,s,'bior3.7',1);
h1=wrcoef2('h',c,s,'bior3.7',1);
v1=wrcoef2('v',c,s,'bior3.7',1);
d1=wrcoef2('d',c,s,'bior3.7',1);
c1=[a1,h1;v1,d1];
%显示分解后各频率成份的信息
subplot(222); image(c1);
axis square
title('分解后低频和高频信息');
%====================================
%下面进行图像压缩处理
%保留小波分解第一层低频信息，进行图像的压缩
%第 1 层的低频信息即为 ca1，显示第 1 层的低频信息
%首先对第 1 层信息进行量化编码
```

```
ca1=appcoef2(c,s,'bior3.7',1);
ca1=wcodemat(ca1,440,'mat',0);
%改变图像的高度
ca1=0.5 * ca1;
subplot(223); image(ca1); colormap(map);
axis square;
title('第一次压缩图像');
disp('第一次压缩图像的大小为:');
whos('ca1')
%--------------------------------------------
%保留小波分解第二层低频信息,进行图像的压缩,此时压缩比更大
%第 2 层的低频信息即为 ca2,显示第 2 层的低频信息
ca2=appcoef2(c,s,'bior3.7',2);
%首先对第 2 层信息进行量化编码
ca2=wcodemat(ca2,440,'mat',0);
%改变图像的高度
ca2=0.25 * ca2;
subplot(224); image(ca2); colormap(map);
axis square;
title('第二次压缩图像');
disp('第二次压缩图像的大小为:');
whos('ca2')
```

输出结果(如图 3.17 所示)。

```
压缩前图像 X 的大小为
    Name        Size          Bytes    Class
    X           256x256       524288   double array

第一次压缩图像的大小为
    Name        Size          Bytes    Class
    ca1         135x135       145800   double array

第二次压缩图像的大小为
    Name        Size          Bytes    Class
    ca2         75x75         45000    double array
```

在这里可以看出，第一次压缩我们是提取原始图像中小波分解第一层的低频信息，此时压缩效果较好，压缩比较小(约为 1/3)；第二次压缩是提取第一层分解低频部分的低频部分(即小波分解第二层的低频部分)，其压缩比较大(约为 1/12)，压缩效果在视觉上也基本过得去。

这种只保留原始图像中低频信息的压缩方法是一种最简单的压缩方法，它不需经过其

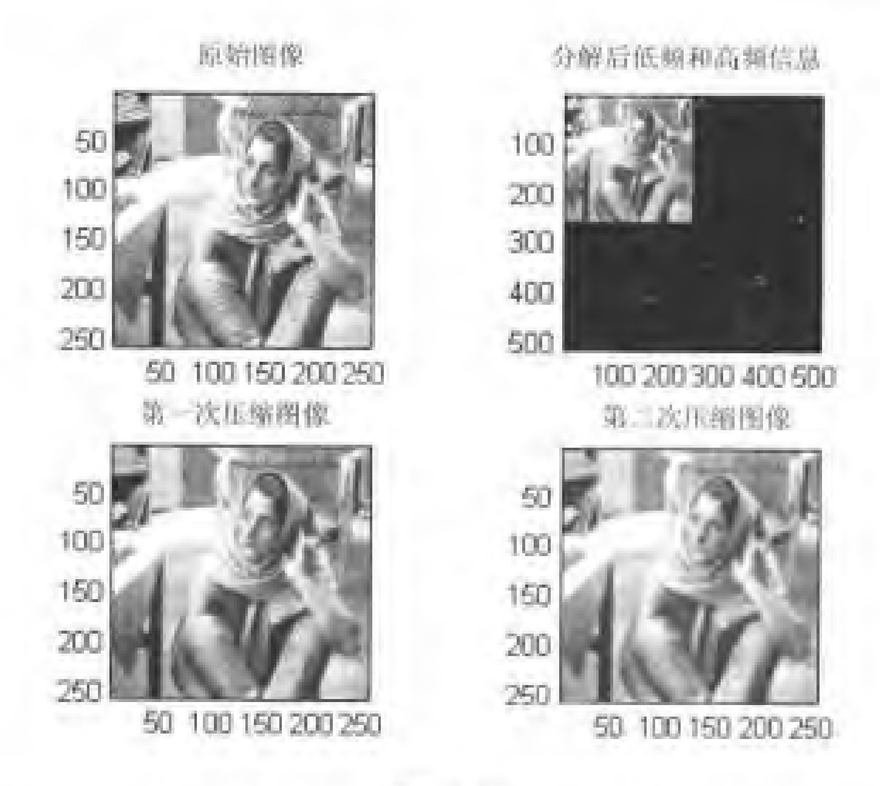


图　3.17

它处理即可获得较好的压缩效果。在上面的例子中，我们还可以只提取小波分解第3、4、…层的低频信息，从理论上说，我们可以获得任意压缩比的压缩图像。由此可以看出，小波分析用于图像压缩具有明显的优点。

例3.17　给出一个轮胎图像(文件名为tire.mat)，请利用wdencmp函数对图像进行压缩。

解：这是一个利用MATLAB中wdencmp函数对图像进行压缩的问题，其处理过程如下：

```
load tire; %装入一个二维信号(图形)
%显示图像
subplot(221); image(X); colormap(map)
title('原始图像');
axis square
%===================================
%下面进行图像压缩处理
% 对图像用db3小波进行2层小波分解
[c,s]=wavedec2(X,2,'db3');
%使用wdencmp命令函数来实现图像的压缩
[thr,sorh,keepapp]=ddencmp('cmp','wv',X);
%在输入参数中，我们选择了全局阈值选项'gbl'，用来对所有高频系数进行相同
%的阈值量化处理
[Xcomp,cxc,lxc,perf0,perfl2]=wdencmp('gbl',c,s,'db3',2,…
```

```
                    thr,sorh,keepapp);
%将压缩后的图像与原始图像相比较，并显示出来
subplot(222); image(Xcomp); colormap(map);
title('压缩后图像');
axis square
disp('小波分解系数中置 0 的系数个数百分比：');
perf0
disp('压缩后图像剩余能量百分比：');
perfl2
```

输出结果(如图 3.18 所示)：

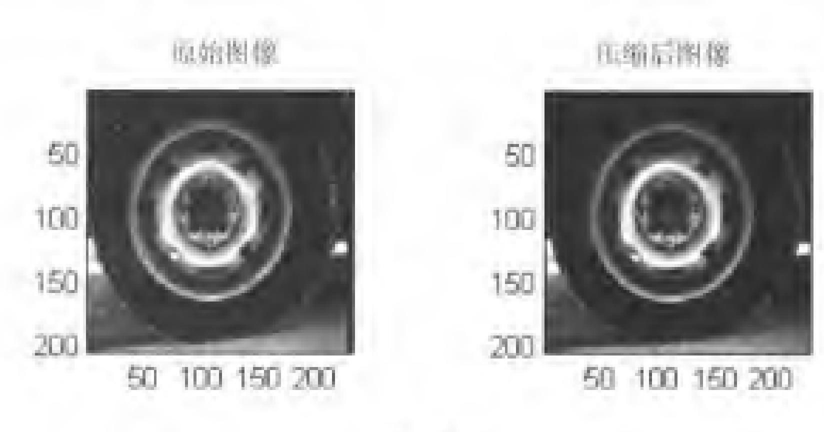


图　3.18

在利用二维小波变换进行图像压缩时需要说明的是：小波变换为图像从空间域变换到时间域提供了一种非常有效的方法，它的作用与以前在图像压缩中所用到的离散余弦变换(DCT)、傅里叶变换等的作用类似。但是，要很好地进行图像的压缩，需要综合地利用多种其它技术，特别是数据编码与解码算法等，所以利用小波分析进行图像压缩往往是借助小波分析和许多其它相关技术共同完成的。

3.2.2　二维小波分析用于图像消噪

1. 图像消噪方法的一般说明

对二维图像信号的消噪方法同样适用于一维信号，尤其是对于几何图像更适合。二维模型可以表述为

$$s(i,j) = f(i,j) + \sigma e(i,j) \qquad i,j = 0,\cdots,m-1$$

其中，e 是标准偏差不变的高斯白噪声。二维信号的消噪步骤与一维信号的消噪步骤完全相同，也有三步，只是用二维小波分析工具代替了一维小波分析工具。如果用固定的阈值形式，则选择的阈值用 m^2 代替了一维信号中的 n。这三步是：

(1) 二维信号的小波分解。选择一个小波和小波分解的层次 N，然后计算信号 s 到第 N 层的分解。

(2) 对高频系数进行阈值量化。对于从 1 到 N 的每一层，选择一个阈值，并对这一层

的高频系数进行软阈值量化处理。

(3) 二维小波的重构。根据小波分解的第 N 层的低频系数和经过修改的从第1层到第 N 层的各层高频系数，来计算二维信号的小波重构。

在这三个步骤中，重点内容就是如何选取阈值和如何进行阈值的量化。请注意，除了一维信号自动消噪的情况，对于其它的情况，一维信号的消噪和压缩用的是wdencmp，这对于二维信号也是一样的。

例3.18 给定一个二维信号(文件名为：sinsin.mat)，请利用小波分析对信号进行消噪处理。

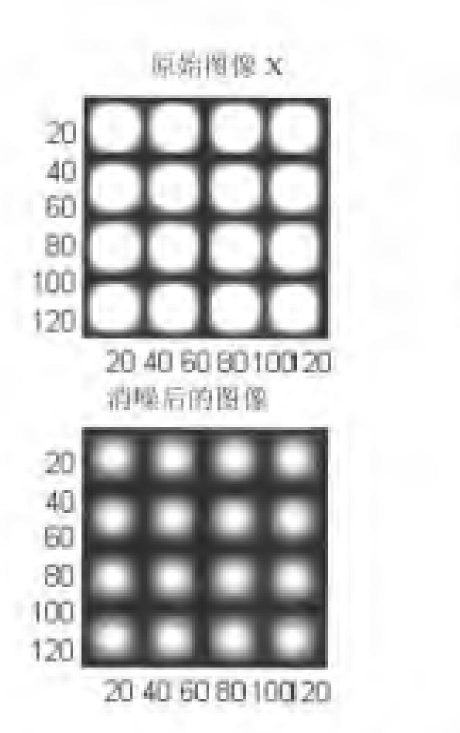
解：该问题是一个用二维小波分析进行信号消噪处理的典型实例。在这里，我们用MATLAB所提供的ddencmp和wdencmp消噪函数实现消噪，其消噪过程可以按照如下程序进行。

程序清单：

```
%装入名为sinsin的二维图像
load sinsin;
%下面进行噪声的产生
init=2788605826;
rand('seed',init); Xnoise=X+18 * (rand(size(X)));
%显示原始图像及它的含噪声的图像
colormap(map);
subplot(2,2,1); image(wcodemat(X,192));
title('原始图像X')
axis square
subplot(2,2,2); image(wcodemat(Xnoise,192));
title('含噪声的图像Xnoise')
axis square
%==================================================
%用sym5小波对图像信号进行二层的小波分解
[c,s]=wavedec2(X,2,'sym5');
%下面进行图像的消噪处理
%使用ddencmp函数来计算消噪的默认阈值和熵标准
%使用wdencmp函数来实现图像的压缩
[thr,sorh,keepapp]=ddencmp('den','wv',Xnoise);
[Xdenoise,cxc,lxc,perf0,perfl2]=wdencmp('gbl',c,s,'sym5',2, ...
                            thr,sorh,keepapp);
%显示消噪后的图像
subplot(223); image(Xdenoise);
title('消噪后的图像'); axis square
```

输出结果(如图3.19所示)。

从下面三个图像的比较可以看出，MATLAB中的ddencmp和wdencmp函数可以有效



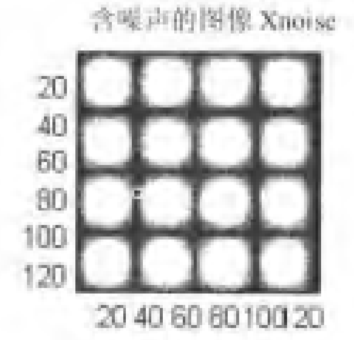


图　3.19

地进行消噪处理。

例 3.19　给定一个有较大白噪声的 tire.mat 图像，利用二维小波变换对图像进行消噪。

解：由于图像所含的噪声主要是白噪声，而且主要集中在图像的高频部分，所以我们可以通过全部滤掉图像中的高频部分实现图像的消噪。具体消噪过程可由如下程序进行：

```
%下面装载原始图像，X 中含有被装载的图像
load tire;
%画出原始图像
subplot(221); image(X); colormap(map);
title('原始图像');
axis square
%产生含噪图像
init=2055615866; randn('seed',init)
x=X+38*randn(size(X));
%画出含噪声图像
subplot(222); image(x); colormap(map);
title('含噪声图像');
axis square;
%==================================
%下面进行图像的消噪处理
%用小波函数 sym4 对 x 进行 2 层小波分解
```

```
[c,s]=wavedec2(x,2,'sym4');
%提取小波分解中第一层的低频图像,即实现了低通滤波消噪
a1=wrcoef2('a',c,s,'sym4',1);
%画出消噪后的图像
subplot(223); image(a1);
title('第一次消噪图像');
axis square;
%提取小波分解中第二层的低频图像,即实现了低通滤波消噪
%这相当把第一层的低频图像经过再一次的低频滤波处理
a2=wrcoef2('a',c,s,'sym4',2);
%画出消噪后的图像
subplot(224); image(a2);
title('第二次消噪图像');
axis square;
```

输出结果(如图3.20所示):

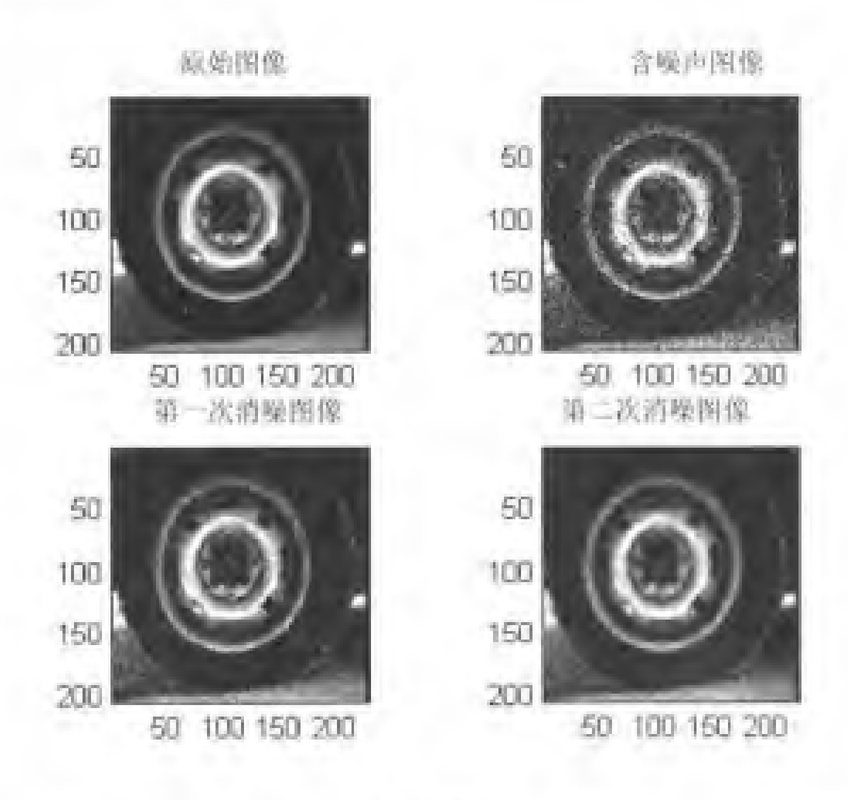


图 3.20

从上面的输出结果可以看出，第一次消噪已经滤去了大部分的高频噪声，但从消噪图像与原始图像相比可以看出，第一次消噪后的图像中还是含有不少的高频噪声；第二次消噪是在第一次消噪的基础上，再次滤去其中的高频噪声，从消噪的结果可以看出，它具有较好的消噪效果。

例3.20　给定一个只含有较少噪声的facets.mat图像，试用二维小波分析进行图像消

噪处理。

解：对于该问题，由于原始图像中只含有较少的高频噪声，如果采用例 3.19 中把高频噪声全部滤波的方法将会损害图像中固有的高频有用信号。在这里，我们采用小波分解系数阈值量化的方法进行消噪处理。

```
%下面装载原始图像，X 中含有被装载的图像
load facets;
%画出原始图像
subplot(221); image(X); colormap(map);
title('原始图像');
axis square
%产生含噪图像
init=2055615866; randn('seed',init)
x=X+10*randn(size(X));
%画出含噪声图像
subplot(222); image(x); colormap(map);
title('含噪声图像');
axis square;
%=================================
%下面进行图像的消噪处理
%用小波函数 coif3 对 x 进行 2 层小波分解
[c,s]=wavedec2(x,2,'coif3');
%提取小波分解中第一层的低频图像,即实现了低通滤波消噪
%设置尺度向量 n
n=[1,2];
%设置阈值向量 p
p=[10.12,23.28];
%对三个方向高频系数进行阈值处理
nc=wthcoef2('h',c,s,n,p,'s');
nc=wthcoef2('v',c,s,n,p,'s');
nc=wthcoef2('d',c,s,n,p,'s');
%对新的小波分解结构[nc,s]进行重构
xx=waverec2(nc,s,'coif3');
%画出重构后图像的波形
subplot(223); image(xx);
title('消噪后的图像');
axis square;
```

输出结果(如图 3.21 所示):

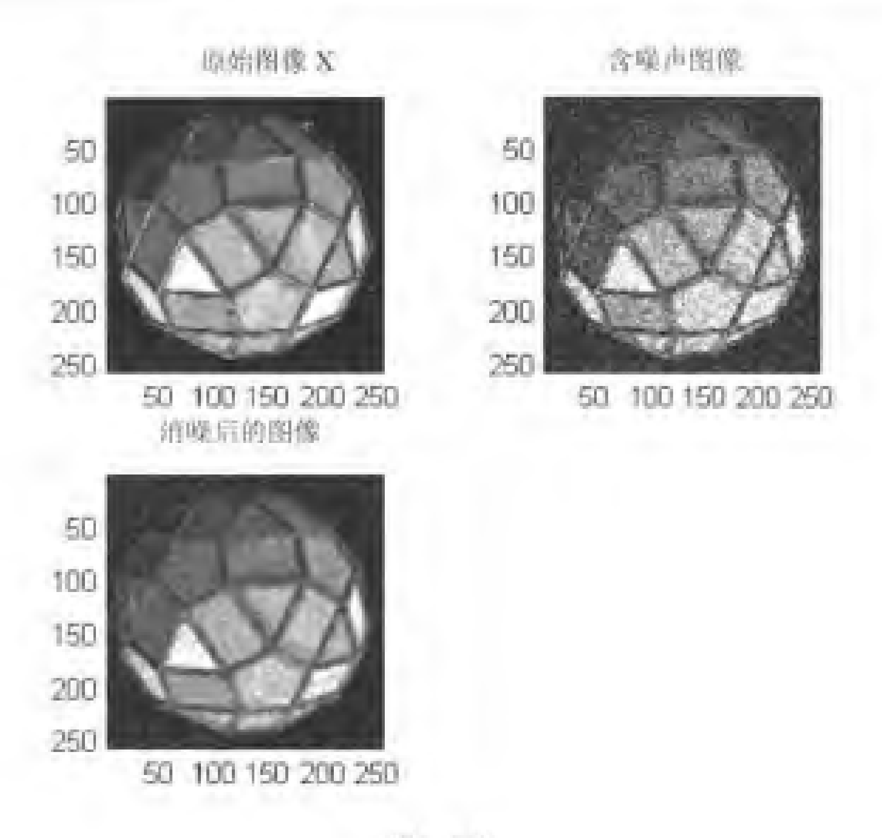


图 3.21

3.2.3 二维小波分析的其它应用

1. 图像增强

小波变换将一幅图像分解为大小、位置和方向都不同的分量。在做逆变换之前可以改变小波变换域中某些系数的大小，这样就能够有选择地放大所感兴趣的分量而减小不需要的分量。下面给出一个图像增强的实例。

例 3.21　给定一个 woman.mat 信号，试用二维小波分析对图像进行增强处理。

解：由于图像经二维小波分解后，图像的轮廓主要体现在低频部分，而细节部分则体现在高频部分，因此，可以通过对低频分解系数进行增强处理，对高频分解系数进行衰减处理，即可以达到图像增强的作用。具体处理过程见如下程序：

```
load woman;
%画出原始图像
subplot(221); image(X); colormap(map);
title('原始图像');
axis square
%==================================
%下面进行图像的增强处理
%用小波函数 sym4 对 X 进行 2 层小波分解
[c,s]=wavedec2(X,2,'sym4');
```

```
sizec=size(c);
%对分解系数进行处理，通过处理，突出轮廓部分，弱化细节部分
for i=1:sizec(2)
        if(c(i)>350)
           c(i)=2*c(i);
        else
           c(i)=0.5*c(i);
        end
end
%下面对处理后的系数进行重构
xx=waverec2(c,s,'sym4');
%画出重构后的图像
subplot(222); image(xx);
title('增强图像');
axis square;
```

输出结果(如图 3.22 所示)：





图　3.22

2. 图像融合

图像融合是将同一对象的两个或更多的图像合成在一幅图像中，以便它比原来的任何一幅更能容易地为人们所理解。这一技术可应用于多频谱图像理解以及医学图像处理等领域，在这些场合，同一物体部件的图像往往是采用不同的成像机理得到的。

例 3.22　试用二维小波分析将 woman.mat 和 wbarb.mat 图像融合在一起。

解：其处理过程如下：

```
load woman;
X1=X; map1=map;
%画出原始图像
subplot(221); image(X1); colormap(map1);
title('woman');
axis square
```

```
%装入图像
load wbarb;
X2=X; map2=map;
for i=1:256
   for j=1:256
        if(X2(i,j)>100)
           X2(i,j)=1.2*X2(i,j);
        else
           X2(i,j)=0.5*X2(i,j);
        end

   end
end
%画出图像
subplot(222); image(X2); colormap(map2);
title('wbarb');
axis square
%==================================
%用小波函数sym4对X1进行2层小波分解
[c1,s1]=wavedec2(X1,2,'sym4');
sizec1=size(c1);
%对分解系数进行处理，通过处理，突出轮廓部分，弱化细节部分
for i=1:sizec1(2)
        c1(i)=1.2*c1(i);
end
%用小波函数sym4对X2进行2层小波分解
[c2,s2]=wavedec2(X2,2,'sym4');
%下面进行小波变换域里的图像融合
c=c1+c2;
%减小图像的亮度
c=0.5*c;
%对融合的系数进行重构
xx=waverec2(c,s1,'sym4');
%画出重构后的图像
subplot(223); image(xx);
title('融合图像');
axis square;
```

输出结果(如图3.23所示)。

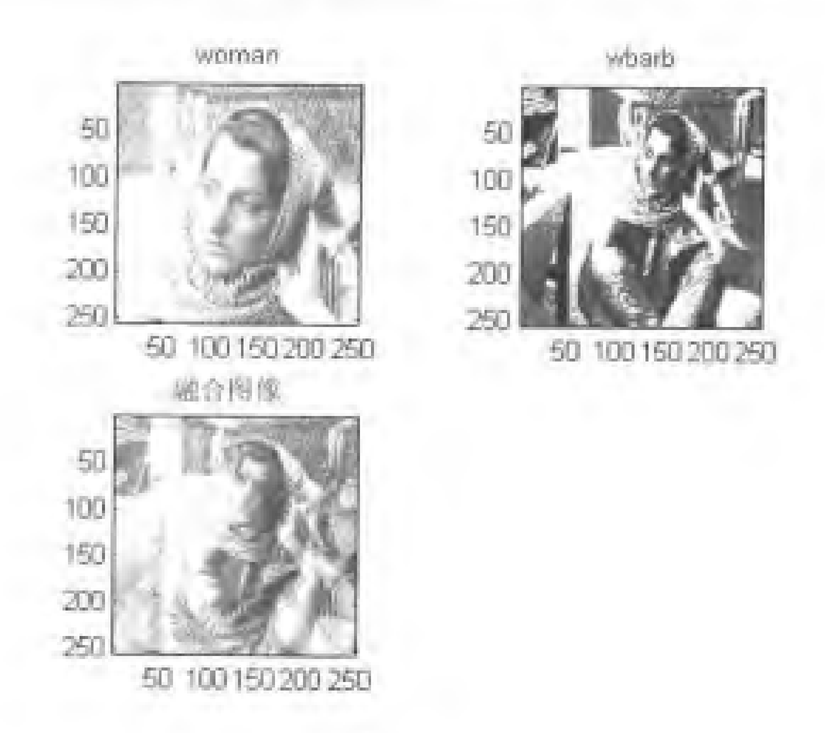


图 3.23

3. 图像平滑处理

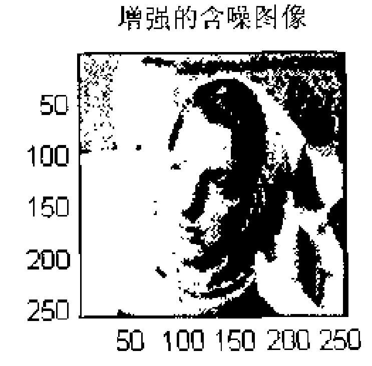
例 3.23　给定一个含噪声的 woman 图像，试用二维小波分析和图像的中值滤波进行图像的平滑。

解：该问题是一个图像平滑处理问题。首先，对图像在频域内进行增强，然后在空间域内加入较大的白噪声。通过对含噪图像进行平滑处理，即可以使含噪图像具有较好的平滑效果。其具体处理过程如以下程序：

```
%装入原始信号
load woman;
%用 sym4 小波对图像进行 2 层分解
[c,s]=wavedec2(X,2,'sym4');
sizec=size(c);
%在频域里对图像进行增强
for i=1:sizec(2)
        if(c(i)>350)
            c(i)=1.3*c(i);
        else
            c(i)=0.5*c(i);
        end
end
%对经过增强处理后的系数进行重构
```

```
xx=waverec2(c,s,'sym4');
%加入噪声
init=2788605826;
rand('seed',init); xx=xx+68*(rand(size(xx)));
subplot(221); image(xx);
title('增强的含噪图像');
axis square;
%=====================================
%下面对图像进行中值滤波处理
for i=2:1:255
   for j=2:1:255
      temp=0;
      for m=1:3
         for n=1:3
            temp=temp+xx(i+m-2,j+n-2);
         end
      end
      temp=temp/9;
      xx(i,j)=temp;
   end
end
%画出平滑后的图像
colormap(map);
subplot(222);image(xx);axis square
title('平滑后的图像');
```

输出结果(如图 3.24 所示):



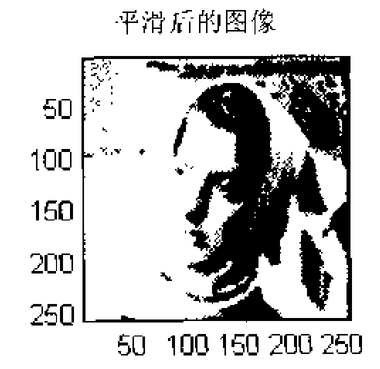


图 3.24

3.3　小波包分析的应用

小波分析是把信号 s 分解成低频 a1 和高频 d1 两部分，在分解中，低频 a1 中失去的信息由高频 d1 捕获。在下一层的分解中，又将 a1 分解成低频 a2 和高频 d2 两部分，低频 a2 中失去的信息由高频 d2 捕获，如此类推下去，可以进行更深层次的分解；小波包分解则不然，它不仅对低频部分进行分解，而且还对高频部分也进行分解(参见第 1 章的第 1.5 节)。因此，小波包分解是一种比小波分解更为精细的分解方法。正因为如此，小波包分析是一种更广泛应用的小波分解方法，它广泛应用于各种信号处理，包括信号的分解、编码、消噪、压缩等等。下面就小波包对信号的分解特性和它在信号处理中的应用分别介绍。

首先，我们将用于小波包分析的主要函数作以简要介绍，这些函数在第 2 章中已做过详细说明，在此，只是为了方便读者的使用而作一个归纳总结，具体每个函数的用法，请参阅第 2 章的有关内容。用于小波包分析的函数，主要有：

(1) 小波包分解函数

表 3 - 12　小波包分解函数

函数名	功　能
wpdec	一维小波包分解
wpdec2	二维小波包分解
wpsplt	分割(分解)小波包

(2) 小波包重构函数

表 3 - 13　小波包重构函数

函数名	功　能
wprcoef	小波包分解系数的重构(一维或二维)
wprec	一维小波包分解的重构
wprec2	二维小波包分解的重构
wpjoin	重新组合小波包

(3) 小波分解结构操作函数

表 3 - 14　小波分解结构操作函数

函数名	功　能
wpcutree	剪切小波包分解树
bestree	计算最佳(优)树
bestlevt	计算完整最佳小波包树
wentropy	计算小波包的熵
entrupd	更新小波包的熵值

(4) 小波包消噪和压缩函数

表 3-15 小波包消噪和压缩函数

函数名	功　能
wthresh	进行软阈值或硬阈值处理
ddencmp	求解消噪或压缩过程中的默认值阈值(软或硬)、熵标准
wpthcoef	进行小波包分解系数的阈值处理
wpdencmp	用小波包进行信号的消噪或压缩

3.3.1 小波包的构造

当用一个正交小波去构造一个小波包时，计算方法很简单。首先，与所选择的小波相对应的两个长为 $2N$ 的滤波器($h(n)$和 $g(n)$)，它们分别是低通分解滤波器和高通分解滤波器的被 $\sqrt{2}$ 除过之后的重构滤波器。在详细介绍之前，先定义下面的函数序列($W_n(x)$，$n=0,1,2,\cdots$)：

$$W_{2n} = 2\sum_{k=0}^{2N-1} h(k)W_n(2x-k)$$

$$W_{2n+1} = 2\sum_{k=0}^{2N-1} g(k)W_n(2x-k)$$

其中，$W_0(x)=\phi(x)$是尺度函数，$W_1(x)=\Psi(x)$是小波函数。

例如，对 haar 小波函数：

$$h(0) = h(1) = 1/2 \qquad N = 1$$

$$g(0) = -g(1) = 1/2 \qquad N = 1$$

等式变为

$$W_{2n}(x) = W_n(2x) + W_n(2x-1),$$

$$W_{2n+1}(x) = W_n(2x) - W_n(2x-1)$$

在这里，$W_0(x)=\phi(x)$是 Haar 尺度函数，$W_1(x)=\Psi(x)$是 Haar 小波函数，两个函数的支撑长度均在区间[0,1]上。从上式我们可以看出，可以通过把支撑区间分别在[0,1/2]和[1/2,1]内的两个 1/2 尺度的 W_n 加起来获得 W_{2n}函数。同样，可以通过把支撑区间分别在[0,1/2]和[1/2,1]内的两个 1/2 尺度的 W_n 相减获得 W_{2n+1}函数。

例 3.24 分别构造 Haar 和 db2 小波函数的小波包(n 从 0 到 7)，并画出其图形。

解：这是一个典型的小波包构造的实例，它可以通过 wpfun 函数完成，具体的构造过程可按如下程序实现。

程序清单：

```
[wfun,xgrid]=wpfun('haar',7,5); %n=0,1, …7
figure(1);
for i=1:8
    w=wfun(i,:);
    subplot(4,2,i); plot(w);
    Ylabel(['W',num2str(i)]);
end
```

```
[wfun,xgrid]=wpfun('db2',7,5); %n=0,1, …7
figure(2);
for i=1:8
    w=wfun(i,:);
    subplot(4,2,i); plot(w);
    Ylabel(['W',num2str(i)]);
end
```

输出结果(如图 3.25 所示):

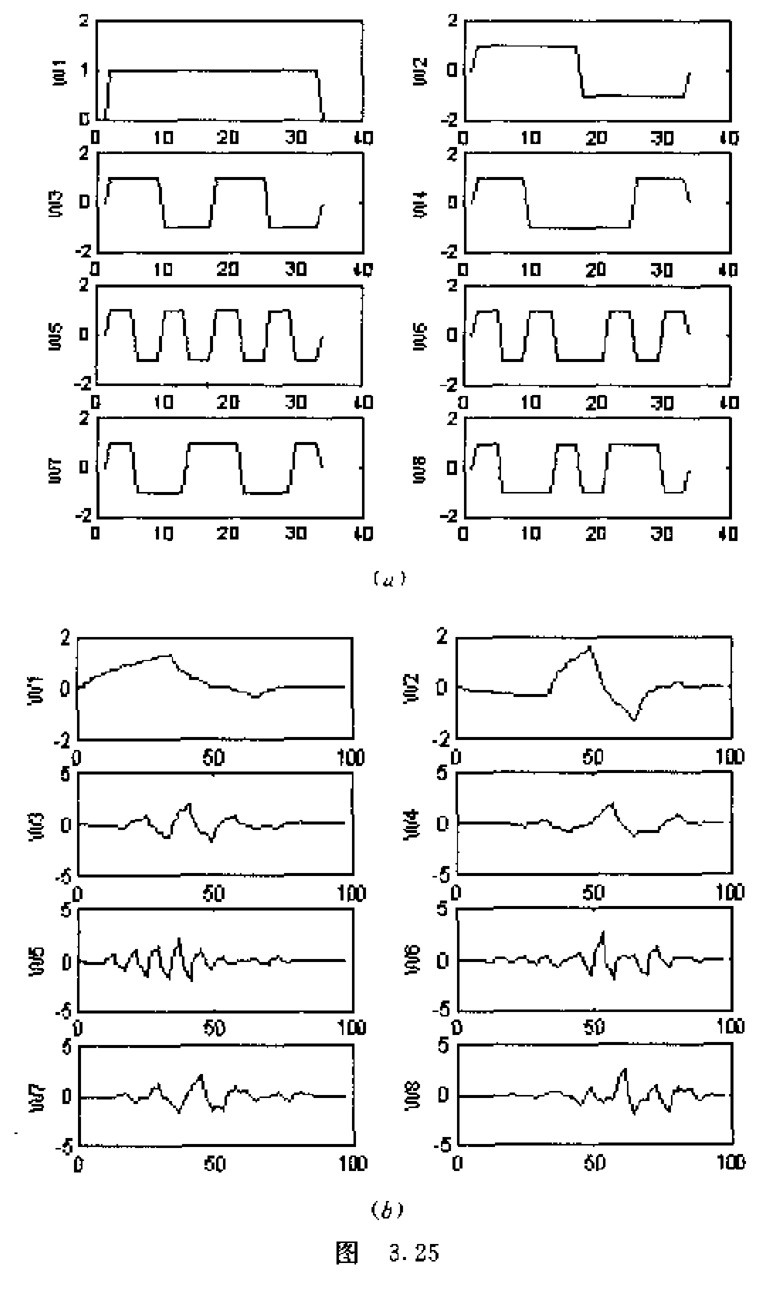


图　3.25

3.3.2　小波包的元素

从函数 $W_n(x)(n=0,1,2,\cdots)$和相应的正交小波包函数，我们可以得到用三个参数描

述的小波包函数

$$W_{j,n,k}(x)=2^{-j/2}W_n(2^{-j}x-k),\qquad n\in N;\ (j,k)\in Z^2$$

在小波标架中，k 是一个时间位置的参数，j 是一个小波尺度参数，而在这里，n 代表什么呢？从图 3.25 中我们可以看到，函数 $W_n(x)$ 大约震荡 n 次。所以对于固定的 j 和 k，$W_{j,k,n}$ 分析的是信号在时间位置 $2^j k$ 附近、小波尺度 2^{-j} 下的振荡情况。

实际上，如果仔细观察小波包的图形，对于按自然数增加的顺序排列的 $W_n(x)$，即 $n=0,1,\cdots,7,\cdots$，它们震荡的次数不是严格按照自然数的递增而排列的。例如数 db1 小波包的过零点个数，我们可以得到表 3-16 的过零点数据。

表 3-16

$W_n(x)$ 函数中 n 的顺序	0	1	2	3	4	5	6	7
db1 小波包 $W_n(x)$ 的过零点个数	2	3	5	4	9	8	6	7

为了保持主要频率随 n 的增加而单调递增的性质，我们需要对获得的过零点个数顺序进行重新定义，即定义"频率"顺序。它的顺序如表 3-17。

表 3-17

$W_n(x)$ 函数中 n 的顺序	0	1	2	3	4	5	6	7
"频率"顺序 $\sigma(n)$	0	1	3	2	6	7	5	4

在分析一个信号时，最好按照"频率"递增的顺序画出小波包系数的图形，而不是自然生成的顺序，画图时，低频在下方，高频在上方。

3.3.3 小波包的组织管理

函数集 $W_{j,n}=(W_{j,n,k}(x),\ k\in Z)$ 是一个 (j,n) 小波包，对正数 j 和 n，小波包以树结构来组织。下图中树结构的最大分解层等于 3，对每一个小波包分解尺度 j，n 可以取 $0,1,\cdots,2j-1$ 这些数值。图 3.26 显示的是各个 W 函数在小波包分解结构中的位置。

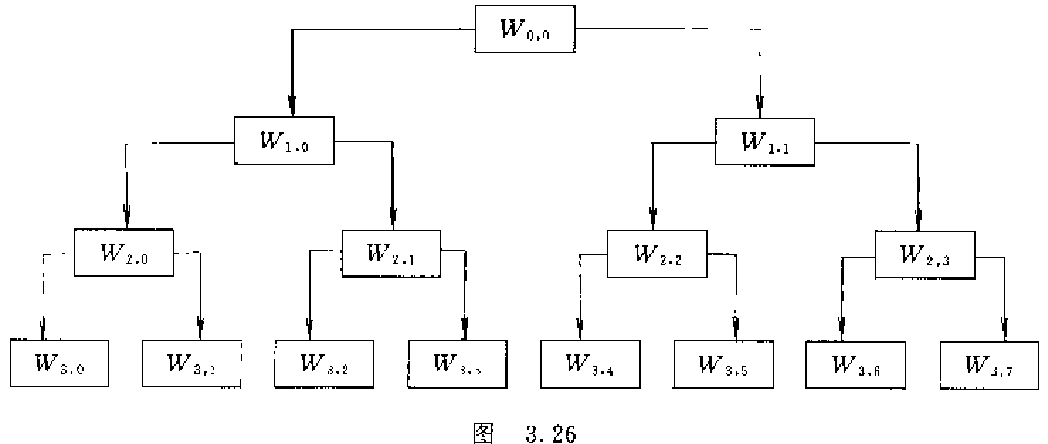


图 3.26

在小波包用树结构来组织时，j 是一个尺度参数，定义的是小波包分解的深度；n 是一个频率参数，定义的是函数 $W_n(x)$ 在树结构中的位置。其中

$$W_{0,0} = (\phi(x-k), k \in Z)$$
$$W_{1,1} = (\Psi(x-k), k \in Z)$$

可以证明，在小波包基函数系列中，包含了小波基函数。进一步分析可得：如果以 V_0 代表所分析的信号所在的空间(由 $W_{0,0}$张成)，则$(W_{d,l}, d)\geqslant l$ 是空间 V_0 的一组正交基。对每一个正整数 $D(W_{D,0}, (W_{d,l}, l\leqslant d\leqslant D))$，也是空间 V_0 的一个正交基。并且$\{(W_{j-1,2n}), (W_{j-1,2n+1})\}$是由 $W_{j,n}$所张成的空间的正交基。这种性质给出了小波包结构树分割的最精确的解释，因为所有的结点都具有图 3.27 的形式。

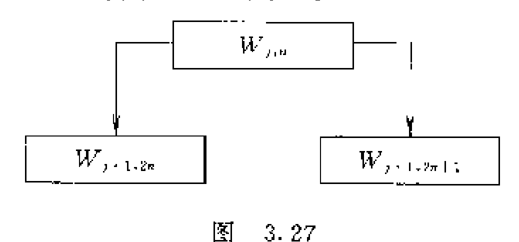


图　3.27

由此可以得出：每一个小波包树的二叉子树都对应着最初的基空间。对一个能量有限的信号，小波包基可以利用在各个频率子带上的信息，提供一种特定的信号编码和信号重构的方法。

3.3.4　最佳小波包基的选择

在小波包函数库建好之后，对于一个给定的正交小波基，我们很自然地要讨论信号的分解问题。总的说来，一个长度为 $N=2^L$ 的信号最多有 2^N 种不同的分解方法，同时，一个深度为 L 的完全二叉子树的二叉子树的个数也为 2^N。由于这个数字是很大的，对每一种情况进行一一列举也是难以想象的，而我们感兴趣的只是，对于某个标准，能找到一种最优的信号分解方法和一种有效的算法即可，这就是我们下面要讲解的最小熵标准。

可以证明，可加性非常有利于进行二叉子树的高效搜索和它的基本分割。传统的基于熵的标准可以完成这项工作，它可以对给定的信号进行信息相关的性能描述。熵的概念在许多领域中是一个普遍的概念，特别是在信号处理中。下面让我们列举常用的四个熵标准，另外，许多其它的熵标准也可以很容易地组合起来(参见 wentropy)。在下面的叙述中，用 s 代表信号，用 s_i 代表信号 s 在一个正交小波包基上的投影系数。熵 E 必须是一个递增的价值函数，即 $E(0)=0$，$E(s) = \sum_i E(s_i)$ 。

(1) Shannon 熵：

$E1(s_i) = -s_i^2\log(s_i^2)$　所以有

$E1(s) = -\sum_i s_i^2\log(s_i^2)$　约定 $0\log(0)=0$。

(2) l^P 范数$(1\leqslant P<2)$：

$E2(s_i) = |s_i|^P$　所以有

$E2(s) = \sum_i |s_i|^P = \| s \|_P^P$

(3) "对数能量"(log energy)熵：

$E3(s_i) = \log(s_i^2)$　所以有

$E3(s) = \sum_i \log(s_i^2)$ 约定 $\log(0)=0$。

(4) 阈值熵：

如果$|s_i|>\varepsilon$，则 $E4(s_i)=1$；其余，$E4(s_i)=0$。定义 $E4(s) = \sum E4(s_i)$，所以有 $E4(s)$为信号大于阈值 ε 的时间点的个数。

这些熵值可以用 wentropy 函数计算得到。下面我们举两个例子进行说明。

例 3.25 对一个能量为 1 的信号，分别用不同的熵标准来计算其熵值。

解：该问题可以如下处理。

程序清单：

```
s=ones(1,16)*0.25;       %产生一个能量为 1 的常数信号
e1=wentropy(s,'shannon')      %s 计算信号 s 的 Shannon 熵
e2=wentropy(s,'norm',1.5)      %s 计算 s 的 l^1.5熵，它等于(s,1.5)^1.5的范数
e3=wentropy(s,'log energy')      %s 计算信号 s 的对数熵
e4=wentropy(s,'threshold',0.24)      %s 计算信号 s 的阈值熵，阈值为 0.24
```

输出结果：

```
e1 =
    2.7726
e2 =
    2.0000
e3 =
  -44.3614
e4 =
    16
```

例 3.26 用基于 minimum 熵标准的最优小波包来分解一个信号。

解：该问题的处理可如下进行。

程序清单：

```
%产生一个能量为 1 的常数信号，
w00=ones(1,16)*0.25;
e1=wentropy(w00,'shannon') %s 计算信号 w00 的 Shannon 熵
[w10,w11]=dwt(w00,'db1'); %用 haar 小波分割信号 w00
e10=wentropy(w10,'shannon') %计算第一层低频的熵值
e11=wentropy(w11,'shannon') %计算第一层高频的熵值
%输出结果：
e1=
    2.7726
e10=
    2.0794
e11=
```

```
    0
%在这里，w11 是零向量，熵值 e11＝0。因为一个信号分割的递增性由
%e10+e11＝2.0794 给定，与原始信号的熵值(e00＝2.7726)相比，
%e10+e11<e00，分割后熵值减小，这是我们感兴趣的地方。
%现在只分割 w10,因为 w11 是零向量,熵为零,分割无意义
[w20,w21]=dwt(w10,'db1');
e20=wentropy(w20,'sbannon') %计算第二层低频的熵值
e21=wentropy(w21,'shannon') %计算第二层高频的熵值
%输出结果
e20=
    1.3863
e21 =
     0
%在这里，w21 也是零向量，它的熵值为零。e20+0<e10，熵值仍是减小
[w30,w31]=dwt(w20,'db1');
e30=wentropy(w30,'shannon') %计算第三层低频的熵值
e31=wentropy(w31,'shannon') %计算第三层高频的熵值
%输出结果
e30=
    0.6931
e31 =
     0
%在这里，w31 也是零向量，它的熵值为零。e30+0<e20，熵值仍是减小
[w40,w41]=dwt(w30,'db1')
%输出结果
w40=
    1.0000
w41=
    0
e40=wentropy(w40,'shannon') %计算第四层低频的熵值
%输出结果
e40=
    0
%在最后的分解操作中，我们发现进行信号重构，只要一条信息即可。根据
%shannon 熵的原则，只有第四层的小波基是最好的小波包分解基(零值最优熵，
%因为 e41+e31+e21+e11=0)。
%在寻找到最优小波包基后，以上的这些工作可以由下面的语句来实现
s=ones(1,16) * 0.25;
%用 shannon 熵和 db1 小波进行小波包分解
```

```
[t,d]=wpdec(s,4,'haar','shannon');
```

下面图 3.28 显示的是小波包分解信号后，各结点对应的熵值。

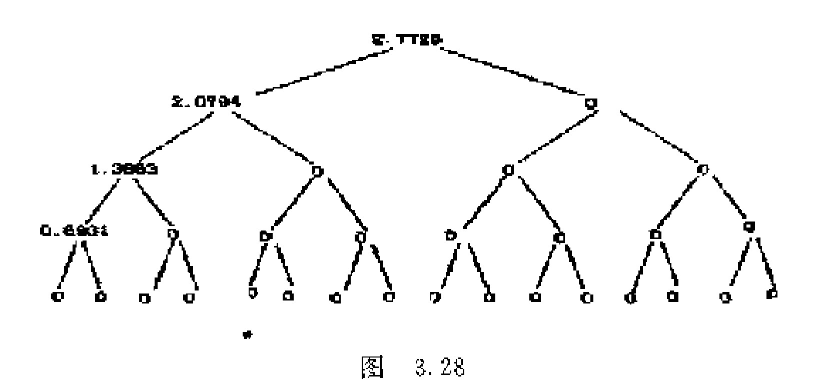


图 3.28

```
[bt,bd]=besttree(t,d); %计算最优小波分解树
```

下面显示的是最优的小波包分解树结构，在这里，最优树就是信号的小波分解树，各个结点上标注的是对应的熵值。

```
%画出最优小波包分解树结构(如图 3.29)。
plottree(bt,bd);
```

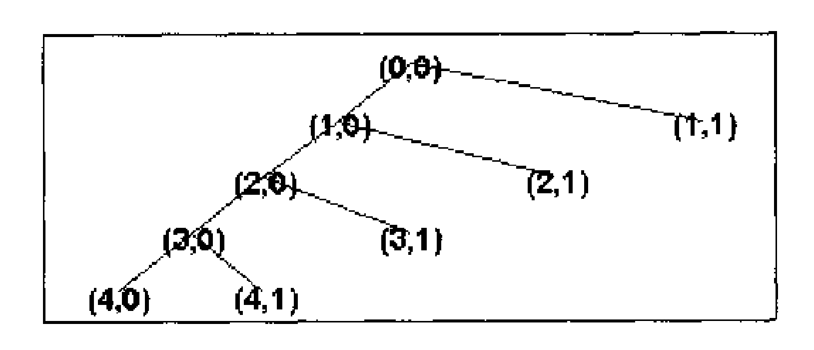


图 3.29

3.3.5 利用小波包进行信号消噪处理

在小波包标架中，其信号消噪和压缩的算法思想与在小波标架中的基本一样，唯一不同的是小波包分析提供了一种更为复杂，同时也更为灵活的分析手段，因为小波包分析对上一层的低频部分和高频部分同时进行细分，具有更为精确的局部分析能力。

对一个信号进行小波包分解，可以采用很多种小波包基，根据所分析信号的要求，从中选择最好的一种小波包基，即最佳基(也叫最优基)，最佳基的选择标准是熵标准。最佳基的选择在 MATLAB 中可用函数 besttree 完成，即计算最佳树。

信号的消噪与压缩是小波包分析的一个最基本的应用，通常，它按照如下几个步骤进行：

(1) 信号的小波包分解。选择一个小波并确定一个小波分解的层次 N，然后对信号 s 进行 N 层小波包分解。

(2) 计算最佳树(即确定最佳小波包基)。对于一个给定的熵标准，计算最佳树，当然，这一步是可选的。对于 GUI 方式，有一个专门的“Best Tree”按钮用于计算最佳树。

（3）小波包分解系数的阈值量化。对于每一个小波包分解系数（特别是低频分解系数），选择一个适当的阈值并对系数进行阈值量化。

（4）小波包重构。根据第 N 层的小波包分解系数和经过量化处理系数，进行小波包重构。

在这四个步骤之中，最关键的就是如何选取阈值和如何进行阈值的量化，从某种程度上说，它直接关系到信号消噪和压缩的质量。

例 3.27　给出一个一维含噪信号，试用小波包分解进行信号的消噪处理。

解：该问题是一个利用小波包进行噪声消除的问题。在这里，首先利用 MATLAB 中所提供的消噪函数 ddencmp 和 wpdencmp 进行默认阈值消噪。然后根据实际工程问题调节阈值的大小，获得更好的消噪效果。

```
%装入原始信号，x 包含装入的信号
load noismima; x=noismima(1:1000);
%画出原始信号
subplot(221); plot(x);
title('原始信号');
%-=================================
%用 wdencmp 进行信号的消噪，并采用默认阈值(参见 ddencmp)
[thr,sorh,keepapp,crit]=ddencmp('den','wp',x)
%用全局阈值选项进行信号的消噪
[xc,treed,datad,perf0,perfl2]= ...
wpdencmp(x,sorh,3,'db2',crit,thr,keepapp);
%画出消噪信号
subplot(222); plot(xc);
title('默认阈值消噪信号');
%根据上面的消噪效果，调节阈值大小进行消噪
thr=thr+15;
[xc1,treed,datad,perf0,perfl2]= ...
        wpdencmp(x,sorh,3,'db2',crit,thr,keepapp);
%画出调节阈值后的消噪信号
subplot(223); plot(xc1);
title('调节后阈值消噪信号');
```

输出结果（如图 3.30 所示）。

例 3.28　给定一个二维图像含噪信号，试用小波包分解进行消噪处理。

解：对于该含噪图像，用小波包进行消噪处理的程序如下：

```
%装入原始图像，X 包含装入的图像
load wgatlin;
%画出原始图像
subplot(221); image(X); colormap(map);
title('原始图像');
```

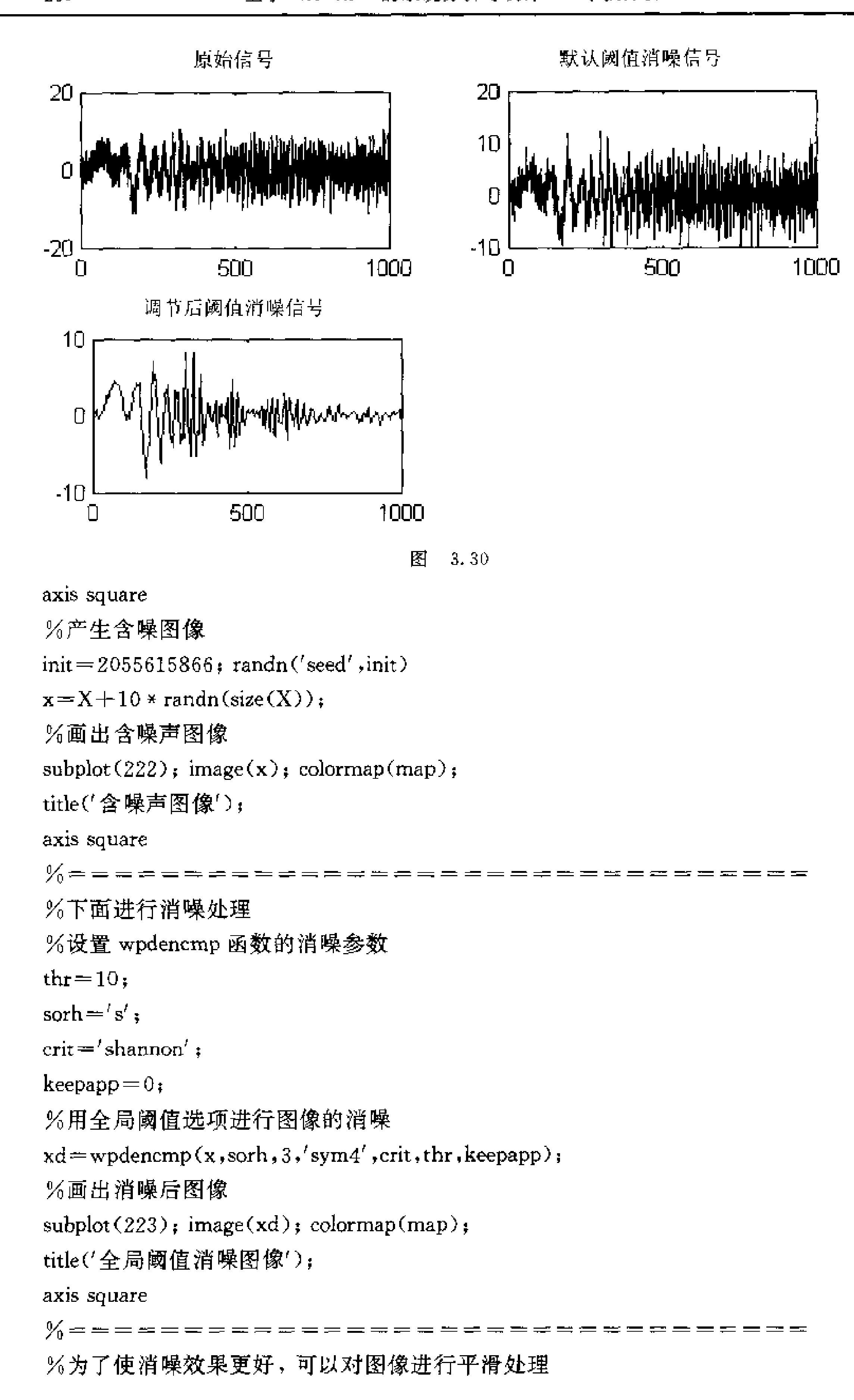


图　3.30

```
axis square
%产生含噪图像
init=2055615866; randn('seed',init)
x=X+10*randn(size(X));
%画出含噪声图像
subplot(222); image(x); colormap(map);
title('含噪声图像');
axis square
%====================================
%下面进行消噪处理
%设置 wpdencmp 函数的消噪参数
thr=10;
sorh='s';
crit='shannon';
keepapp=0;
%用全局阈值选项进行图像的消噪
xd=wpdencmp(x,sorh,3,'sym4',crit,thr,keepapp);
%画出消噪后图像
subplot(223); image(xd); colormap(map);
title('全局阈值消噪图像');
axis square
%====================================
%为了使消噪效果更好，可以对图像进行平滑处理
```

```
%下面对图像进行中值滤波处理
for i=2:1:119
   for j=2:1:179
        temp=0;
        for m=1:3
           for n=1:3
               temp=temp+xd(i+m-2,j+n-2);
           end
        end
        temp=temp/9;
        xd(i,j)=temp;
   end
end
%画出平滑后的图像
subplot(224); image(xd); colormap(map);
title('平滑图像');
axis square
```

输出结果(如图 3.31 所示):

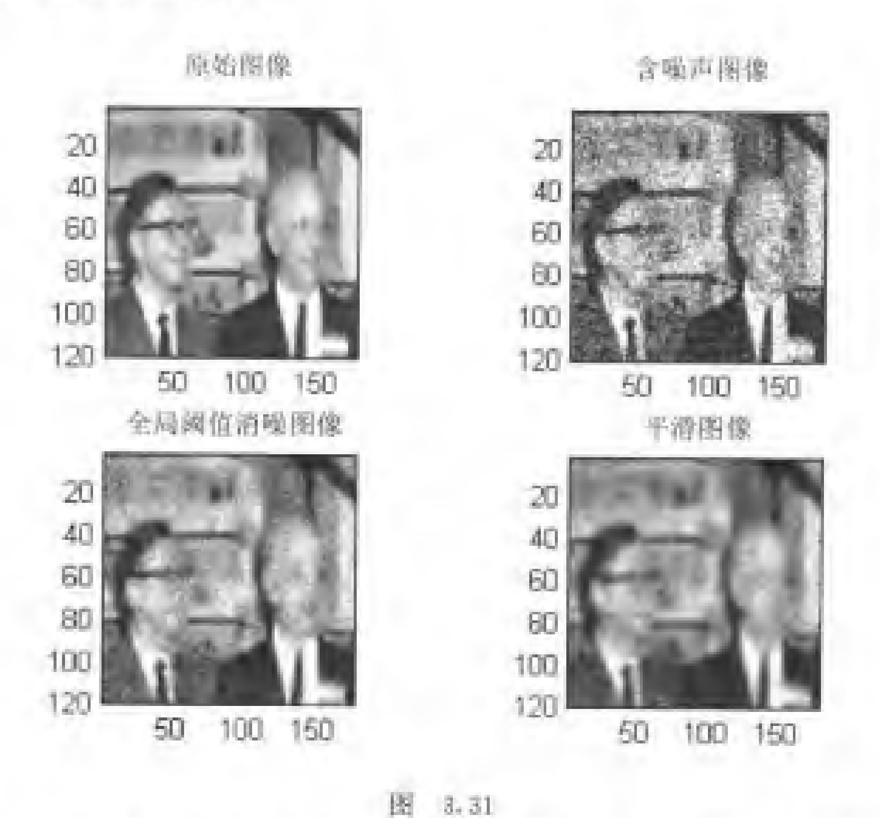


图 3.31

例 3.29 给定一个二维图像含噪信号，试用小波包分解后，设置阈值进行消噪处理。

解：首先产生一个含噪图像，然后用小波包进行图像分解，对它的低频部分进行消噪处理后重构图像。程序如下：

```
%装入一个原始图像
load detfingr;
%画出原始图像
subplot(221); image(X); colormap(map);
title('原始图像');
axis square
init=2055615866; randn('seed',init)
x=X+50*randn(size(X));
%画出含噪声图像
subplot(222); image(x); colormap(map);
title('含噪声图像');
axis square
%用sym2小波对图像进行1层分解
[t1,d1]=wpdec2(x,1,'sym2');
%设置一个全局阈值
thr=10.342;
%对图像分解系数(包括低频分解系数)进行软阈值量化，实现图像消噪
d1=wpthcoef(d1,t1,0,'s',thr);
%由于这里所含噪声主要是高频的白噪声，在此只对低频系数重构
x1=wprcoef(t1,d1,1);
%画出合并重构后的图像
subplot(223); image(x1); colormap(map);
title('消噪后的图像');
axis square
```

输出结果(如图3.32所示)。

3.3.6 利用小波包分析进行图像压缩

例3.30 给出一个一维信号，试用小波包分析对信号进行压缩。

解：在这里，我们采用MATLAB所提供的压缩函数wpdencmp实现一维信号的压缩。具体处理过程如下：

```
%装入原始信号，x包含装入的信号
load noisbump; x=noisbump(1:1000);
%画出原始信号
subplot(221); plot(x);
title('原始信号');
%用wpdencmp进行信号的压缩，并采用默认阈值
[thr,sorh,keepapp,crit]=ddencmp('cmp','wp',x)
%用全局阈值选项进行信号的压缩
[xc,treed,datad,perf0,perfl2]= …
```

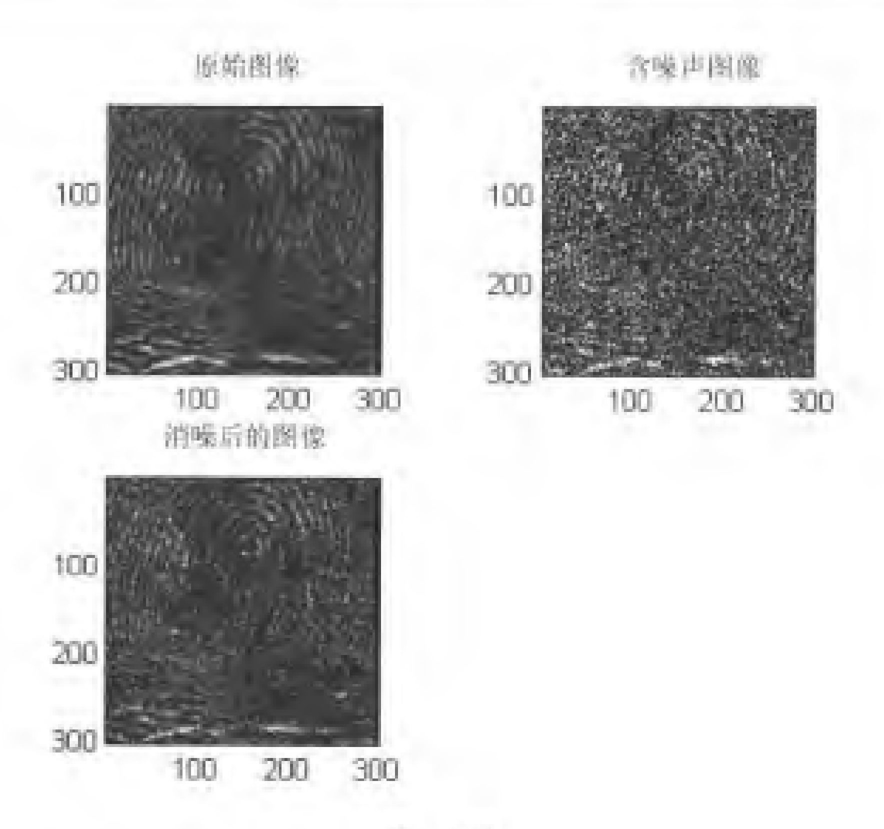


图 3.32

```
wpdencmp(x,sorh,3,'db2',crit,thr,keepapp);
%画出压缩信号
subplot(222); plot(xc);
title('压缩后信号');
```

输出结果(如图 3.33 所示)：

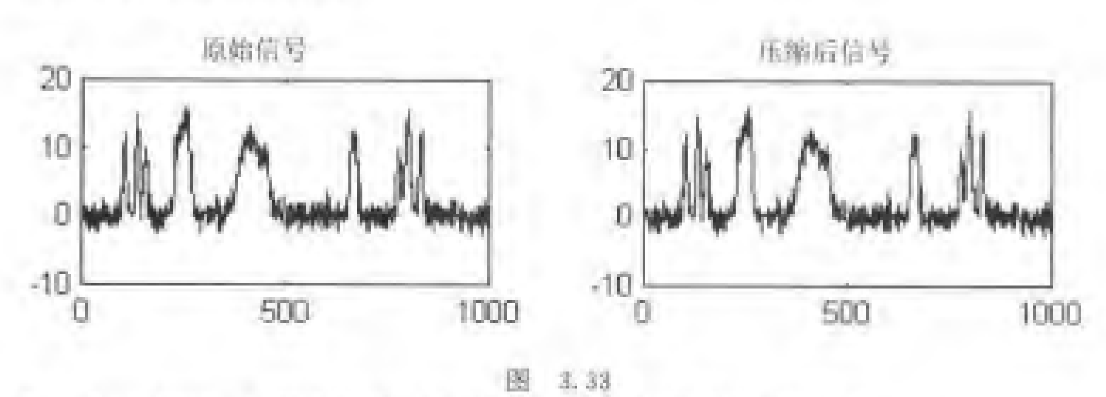


图 3.33

例 3.31　给定一个图像信号，试用小波包分析对图像进行压缩。

解：采用 MATLAB 所提供的压缩函数 wpdencmp 实现二维图像的压缩。具体处理过程如以下程序：

```
%装入原始图像，X 包含装入的图像
load wifs;
%画出原始图像
```

```
subplot(221); image(X); colormap(map);
title('原始图像');
axis square
%用 wpdencmp 进行图像的压缩，采用默认阈值(参见 ddencmp)
[thr,sorh,keepapp,crit]=ddencmp('cmp','wp',X)
%用全局阈值选项进行图像的压缩
xc=wpdencmp(X,sorh,3,'bior3.1',crit,thr,keepapp);
%画出压缩后图像
subplot(222); image(xc); colormap(map);
title('全局阈值压缩图像');
axis square
```

输出结果(如图 3.34 所示)：

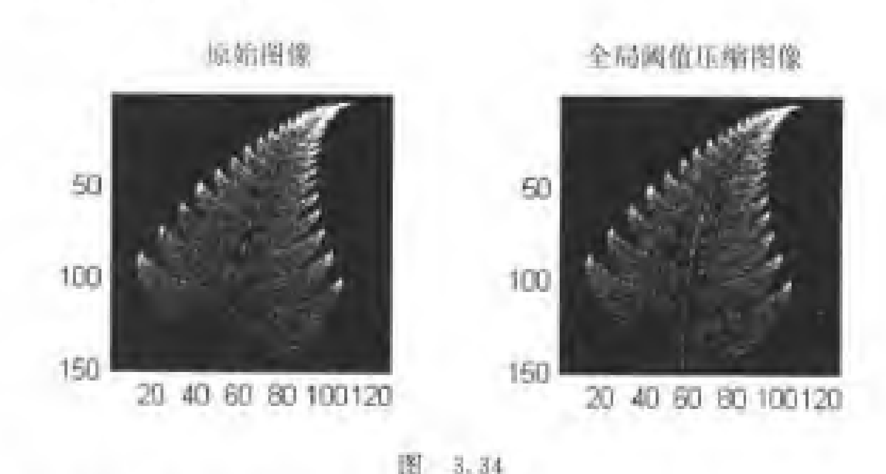


图 3.34

3.3.7 利用小波包分析进行特征提取

例 3.32 现有某导弹控制系统的一级变换放大器(俯仰波道)在正常和故障情况下的大量输出采样数据(以伪随机信号作为输入激励信号)，现要利用小波包分析建立一个能够表征系统状态的特征向量，以便用模式识别的方法对该系统进行故障检测与定位。

解：该问题是一个实际的工程应用问题，它利用小波包分解能够对信号进行精确的细分，提取系统的特征信息，从而实现对系统的故障诊断。问题的解决可按如下方法进行。

1. 变换放大器的频域分析

首先，我们对变换放大器的频域特征进行一下简要分析，当系统出现故障时，其传递函数将会改变，不同频率的幅频特性和相频特性将会有不同程度的改变。对比图 3.35 中的(a)和(b)可看出：当系统出现故障时，其幅频特性和相频特性都有着明显的改变。从幅频特性来说，它主要表现在对不同频率段的输入信号具有不同的抑制和增强作用。

当用一个含有丰富频率成份的信号作为输入对系统进行激励时，由于系统故障对各频率成份的抑制和增强作用发生改变，通常，它会明显地对某些频率成份起着抑制作用，而对另外一些频率成份起着增强作用。因此，其输出与正常系统输出相比，相同频带内信号

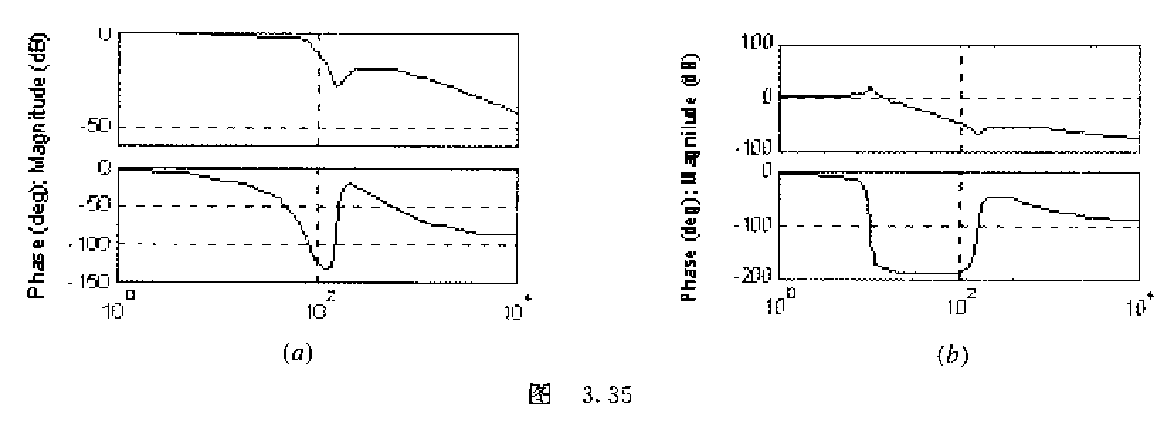


图　3.35

(a) 正常系统幅(相)频特性；(b) 故障系统幅(相)频特性

的能量会有着较大的差别，它使某些频带内信号能量减小，而使另外一些频带内信号能量增大。因此，在各频率成份信号的能量中，包含着丰富的故障信息，某种或某几种频率成份能量的改变即代表了一种故障情况。

从上面的分析可知，输出信号的各频率成份能量的变化表征了系统某些元器件的损坏情况，基于这一点，我们提出了基于"能量—故障"的故障诊断模式识别方法，该方法不需要系统的模型结构，而是直接利用各频率成份能量的变化来诊断故障。利用这一特征就可建立能量变化到物理元器件故障的映射关系，得到表征物理元器件故障的特征向量。选择合适的能量特征化向量对元器件故障进行特征化，可得到每一物理元器件故障的特征向量。上述的原理可总结为下面的算法：

算法：一种提取系统特征信息的小波变模方法

Step1：首先对A/D采样信号进行三层小波包分解，分别提取第三层从低频到高频8个频率成份的信号特征，其分解结构如图3.36所示。

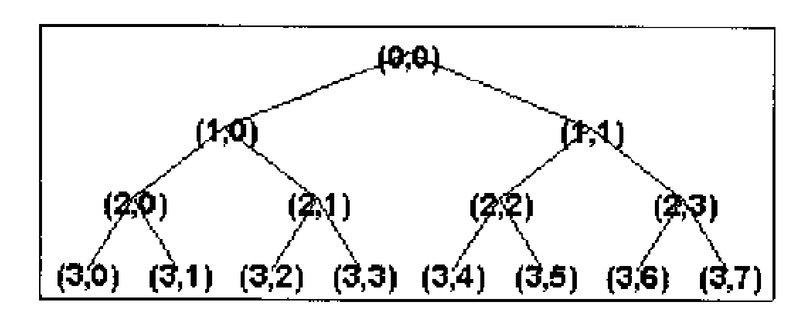


图3.36　小波包三层分解树结构

上图中，(i,j)表示第i层的第j个结点，其中，$i=0,1,2,3$；$j=0,1,\cdots,7$，每个结点都代表一定的信号特征。其中，(0,0)结点代表原始信号S，(1,0)结点代表小波包分解的第一层低频系数X_{10}，(1,1)结点代表小波包分解第一层的高频系数X_{11}，(3,0)结点表示第三层第0个结点的系数，其他依此类推。

Step2：对小波包分解系数重构，提取各频带范围的信号。以S_{30}表示X_{30}的重构信号，S_{31}表示X_{31}的重构信号，其他依此类推。在这里，只对第三层的所有结点进行分析，则总信号S可以表示为

$$S = S_{30} + S_{31} + S_{32} + S_{33} + S_{34} + S_{35} + S_{36} + S_{37}$$

假设原始信号S中，最低频率成份为0，最高频率成份为1，则提取的$S_{3j}(j=0,1,\cdots,$

7)8个频率成份所代表的频率范围见表3-18。

表3-18 各频率成份所代表的频率范围

信号	S_{30}	S_{31}	S_{32}	S_{33}
频率范围	0—0.125	0.125—0.250	0.250—0.375	0.375—0.500
信号	S_{34}	S_{35}	S_{36}	S_{37}
频率范围	0.500—0.625	0.625—0.750	0.750 0.875	0.875—1.000

Step3：求各频带信号的总能量。由于输入信号是一个随机信号，其输出也是一个随机信号。设$S_{3j}(j=0,1,\cdots,7)$对应的能量为$E_{3j}(j=0,1,\cdots,7)$，则有：

$$E_{3j}=\int|S_{3j}(t)|^2dt=\sum_{k=1}^{n}|x_{jk}|^2$$

其中，$x_{jk}(j=0,1,\cdots,7,k=1,2,\cdots,n)$表示重构信号$S_{3j}$的离散点的幅值。

Step4：构造特征向量。由于系统出现故障时，会对各频带内信号的能量有较大的影响，因此，以能量为元素可以构造一个特征向量。特征向量T构造如下：

$$T=[E_{30},E_{31},E_{32},E_{33},E_{34},E_{35},E_{36},E_{37}]$$

当能量较大时，$E_{3j}(j=0,1,\cdots,7)$通常是一个较大的数值，在数据分析上会带来一些不方便的地方。由此，可以对特征向量T进行改进，即对向量进行归一化处理，令

$$E=(\sum_{j=0}^{7}|E_{3j}|^2)^{1/2}$$

$$T'=[E_{30}/E,E_{31}/E,E_{32}/E,E_{33}/E,E_{34}/E,E_{35}/E,E_{36}/E,E_{37}/E]$$

向量T'即为归一化后的向量。

Step5：确定在正常与各种故障状态下，特征向量的特征值及容差范围。特征值和容差范围的确定可以通过机理分析方法求得，同时，也可以通过实验统计的方法确定。机理分析的方法要基于系统的模型，但是，当系统的模型较复杂或根本不知道时，这种方法就显得明显不足；实验统计方法不依赖于系统的数学模型，在工程应用中具有极为重要的意义，这里是以实验统计的方法确定特征值和容差范围。设向量的第一个元素E_{30}/E的特征值为C_0，容差范围是ΔC_0，第二个元素E_{31}/E的特征值为C_1，容差范围为ΔC_1，其他依此类推，即第八个元素E_{37}/E的特征值为C_7，容差范围为ΔC_7。C_j和$\Delta C_j(j=0,1,\cdots,7)$可以通过下式求：

$$C_j=\frac{\sum_{k=1}^{n}x_{jk}}{n}\qquad n\text{为试验次数}$$

如果C_j的值较大时，可对特征值进行归一化处理，令

$$C=\left(\sum_{j=0}^{7}C_j^2\right)^{1/2}$$

归一化的特征向量值为

$$T_{\text{特征值}}=[C_0/C,C_1/C,C_2/C,C_3/C,C_4/C,C_5/C,C_6/C,C_7/C]$$

容差范围$\Delta C_j(j=0,1,\cdots,7)$为

$$\Delta C_j=K\sigma=K\left(\frac{1}{n}\sum_{k=1}^{n}(x_{jk}-C_j)^2\right)^{1/2}\qquad K=3\sim5$$

其中，n 为试验次数，容差范围一般取方差 σ 的 3～5 倍。

如果特征向量的特征值作了归一化处理，则容差范围也应作相应的变化，即容差范围向量的每个元素都相应除以 C，有：

$$\Delta C'_{容差范围} = [\Delta C_0/C,\ \Delta C_1/C,\ \Delta C_2/C,\ \Delta C_3/C,\ \Delta C_4/C,\ \Delta C_5/C,\ \Delta C_6/C,\ \Delta C_7/C]$$

对 n 值的要求：如果实验数据的重复性(或稳定性)较大，则试验次数可以取得较小；如果实验数据的重复性(或稳定性)较小，则要求试验次数 n 较大。

2. 实验设计

以某导弹的一级变换放大器的俯仰波道为研究对象，对上面的分析进行实物验证。实验系统原理如图 3.37 所示：

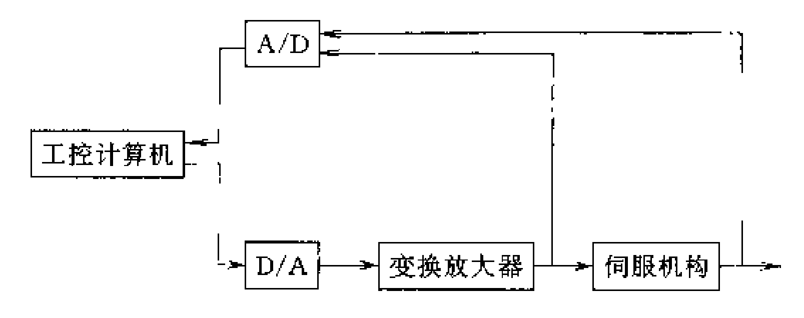


图 3.37　实验系统原理图

系统对输入信号的要求：输入信号要具有丰富的频率成份，能够覆盖系统从低频到高频所有的频率范围。试验中，以伪随机信号(m 序列)作为输入。由于伪随机信号含有丰富的频率成份，所以可以充分地激发系统的各种工作模态，使得系统出现故障时各频率成份能量的变化能得以体现。

3. 实验结果分析

按照所给出的算法，对正常和各种故障状态下的信号进行分析，构造出各自的特征向量，建立"特征向量—系统状态"的对应关系。以一组正常系统和一组故障系统(输入接触不良)的情况为例，其小波包变换后，各频率成份的重构信号如图 3.38。

表 3-19、表 3-20 分别为 5 组正常和故障系统(输入接触不良)测试数据的试验结果。

表 3-19　正常系统测试数据的试验结果表

	E_{30}	E_{31}	E_{32}	E_{33}	E_{34}	E_{35}	E_{36}	E_{37}
第一组	27.2562	19.5412	7.6093	10.4875	3.5911	4.8054	2.1177	3.2799
第二组	27.2399	19.5399	7.6098	10.4900	3.5901	4.8027	2.1256	3.2805
第三组	27.2583	19.5472	7.6086	10.4983	3.5910	4.8045	2.1203	3.2853
第四组	27.2593	19.5518	7.6111	10.4970	3.5916	4.8075	2.1260	3.2814
第五组	27.2676	19.5418	7.6111	10.4818	3.5935	4.8030	2.1179	3.2809

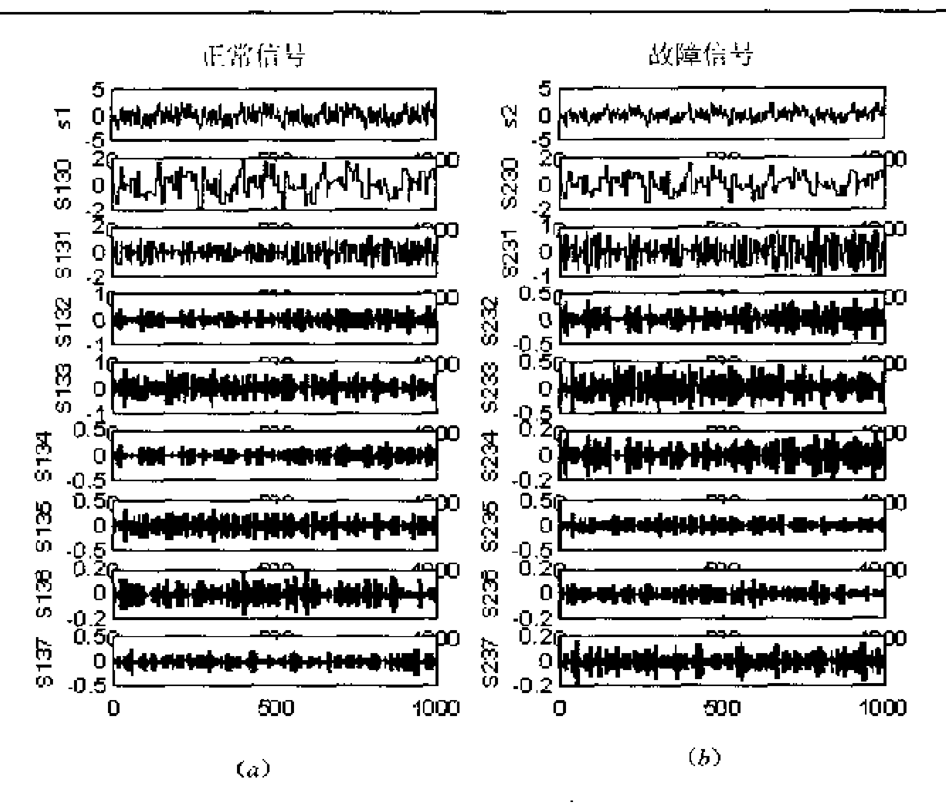


图 3.38　小波包信号特征提取

(a) 正常输出信号及提取的 8 特征量；(b) 正常输出信号及提取的 8 特征量

表 3-20　故障系统(输入接触不良)测试数据的试验结果表

	E_{30}	E_{31}	E_{32}	E_{33}	E_{34}	E_{35}	E_{36}	E_{37}
第一组	20.6384	13.9774	5.4906	7.2795	2.6063	3.3442	1.4371	2.2100
第二组	20.6392	13.9772	5.4911	7.2804	2.6081	3.3443	1.4370	2.2095
第三组	20.6384	13.9780	5.4907	7.2795	2.6064	3.3439	1.4372	2.2097
第四组	20.6391	13.9777	5.4906	7.2798	2.6094	3.3445	1.4376	2.2094
第五组	20.6388	13.9774	5.4908	7.2801	2.6074	3.3440	1.4372	2.2094

从表 3-19 和表 3-20 可以看出，实验数据具有较好的重复性，在确定每种状态的特征向量的特征值和容差范围时，对试验次数 n 要求不需太大，这里取 $n=10$。由于篇幅的限制，每种状态只列出 5 组数据。根据式(12)～(16)，可以求得在正常状态下，特征向量 T 为(注：这里的数据不是很大，所以没有作归一化处理)

$T_{正常}=[27.2563\quad 19.5444\quad 7.6100\quad 10.4909\quad 3.5915\quad 4.8050\quad 2.1215\quad 3.2816]$

容差范围为(取 $K=4$)：

$\Delta C_{正常}=[0.0405\quad 0.0200\quad 0.0044\quad 0.0274\quad 0.0050\quad 0.0069\quad 0.0162\quad 0.0086]$

同样可以求得故障时的特征向量和容差范围：

$T_{故障}=[20.6388\quad 13.9775\quad 5.4908\quad 7.2799\quad 2.6075\quad 3.3442\quad 1.4372\quad 2.2096]$

容差范围为(取 $K=4$)：

$\Delta C_{故障}=[0.0015\quad 0.0013\quad 0.0008\quad 0.0016\quad 0.0051\quad 0.0010\quad 0.0009\quad 0.0010]$

经实验验证：正常情况和输入接触不良时，由测得的数据求出的实际特征向量 98%以

上都与预先所确定的特征向量相一致(在容差范围之内)。依据特征向量进行分类决策，就可以确定是否有故障以及故障的位置。

大量的实验表明：这种基于小波包分析得到的能量特征向量，可以作为故障特征向量进行故障诊断，同时，这种故障诊断方法实施故障特征提取和诊断时，无需被诊断系统的数学模型，就可以迅速地进行故障检测，并能准确地进行故障定位。

附：利用小波包分析进行信号特征提取的源程序：

```
%装入变换放大器输入输出数据
%bf_150ms.dat 为正常系统输出信号
% bf_160ms.dat 为故障系统输出信号
load bf 150ms.dat;
load bf_160ms.dat;
s1=bf_150ms(1:1000); %s1 为正常信号
s2=bf_160ms(1:1000); %s2 为故障信号
%画出正常信号与故障信号的原始波形
title('原始信号');
Ylabel('s1');
subplot(922); plot(s2);
title('故障信号');
Ylabel('s2');
%==================================
%用 db1 小波包对正常信号 s1 进行三层分解
[t,d]=wpdec(s1,3,'db1','shannon');
%plottree(t) %画小波包树结构的图形
%下面对正常信号第三层各系数进行重构
%s130 是指信号 S1 的[3, 0]结点的重构系数，其他依此类推
s130=wprcoef(t,d,[3,0]);
s131=wprcoef(t,d,[3,1]);
s132=wprcoef(t,d,[3,2]);
s133=wprcoef(t,d,[3,3]);
s134=wprcoef(t,d,[3,4]);
s135=wprcoef(t,d,[3,5]);
s136=wprcoef(t,d,[3,6]);
s137=wprcoef(t,d,[3,7]);
%画出重构系数的波形
subplot(9,2,3); plot(s130);
Ylabel('S130');
subplot(9,2,5); plot(s131);
Ylabel('S131');
```

```
subplot(9,2,7); plot(s132);
Ylabel('S132');
subplot(9,2,9); plot(s133);
Ylabel('S133');
subplot(9,2,11); plot(s134);
Ylabel('S134');
subplot(9,2,13); plot(s135);
Ylabel('S135');
subplot(9,2,15); plot(s136);
Ylabel('S136');
subplot(9,2,17); plot(s137);
Ylabel('S137');
%--------------------------------
%计算正常信号各重构系数的方差
%s10是指s130的方差，其他依此类推
s10=norm(s130);
s11=norm(s131);
s12=norm(s132);
s13=norm(s133);
s14=norm(s134);
s15=norm(s135);
s16=norm(s136);
s17=norm(s137);
%向量ss1是针对信号S1构造的向量
disp('正常信号的输出向量')
ss1=[s10,s11,s12,s13,s14,s15,s16,s17]
%================================
%用db1小波包对故障信号s2进行三层分解
[t,d]=wpdec(s2,3,'db1','shannon');
%plottree(t) %画小波包树结构的图形
%下面对故障信号第三层各系数进行重构
%s230是指信号S2的[3,0]结点的重构系数，其他依此类推
s230=wprcoef(t,d,[3,0]);
s231=wprcoef(t,d,[3,1]);
s232=wprcoef(t,d,[3,2]);
s233=wprcoef(t,d,[3,3]);
s234=wprcoef(t,d,[3,4]);
s235=wprcoef(t,d,[3,5]);
s236=wprcoef(t,d,[3,6]);
```

```
s237=wprcoef(t,d,[3,7]);
%画出重构系数的波形
subplot(9,2,4); plot(s230);
Ylabel('S230');
subplot(9,2,6); plot(s231);
Ylabel('S231');
subplot(9,2,8); plot(s232);
Ylabel('S232');
subplot(9,2,10); plot(s233);
Ylabel('S233');
subplot(9,2,12); plot(s234);
Ylabel('S234');
subplot(9,2,14); plot(s235);
Ylabel('S235');
subplot(9,2,16); plot(s236);
Ylabel('S236');
subplot(9,2,18); plot(s237);
Ylabel('S237');
%-----------------------------------------
%计算故障信号各重构系数的方差
%s20 是指 s230 的方差，其他依此类推
s20=norm(s230);
s21=norm(s231);
s22=norm(s232);
s23=norm(s233);
s24=norm(s234);
s25=norm(s235);
s26=norm(s236);
s27=norm(s237);
%向量 ss2 是针对信号 S1 构造的向量
disp('故障信号的输出向量')
ss2=[s20,s21,s22,s23,s24,s25,s26,s27]
```

附录A　MATLAB命令参考

MATLAB系统提供近20类基本命令函数，它们有一部分是MATLAB的内部命令，有一部分是以M文件形式出现的函数，这些M文件按类归于一子目录下，每个目录中除了以M文件表示的函数命令之外，还有一个特殊的文件contents.m，它包含了该目录各个M文件的简介。每个函数文件中都包含了这一函数的用法指南，因此可用命令：

help fn

来显示有关函数fn的帮助信息(fn为M文件名)，也可用命令：

help dn

来显示该目录下各函数文件的简要说明(dn为目录名)。

限于篇幅，本附录不再列出各个函数详细说明，用户可利用help命令获得这些信息，也可参看《MATLAB程序设计语言》一书。

表A.1为20类基本命令函数的子目录及其含义，表A.2～A.20中列出各类函数的简要说明，以供用户参考。

表A.1　基本命令函数目录

目　录　名	命　令　函　数	索　　引
general	通用命令	表A.2
ops	操作符和特殊字符	表A.3
elfun	基本数学函数	表A.4
specfun	特殊数学函数	表A.5
elmat	基本矩阵和矩阵操作	表A.6
specmat	特殊矩阵	表A.7
matfun	矩阵函数 - 数值线性代数	表A.8
sparfun	稀疏矩阵函数	表A.9
datafun	数据分析和傅里叶变换函数	表A.10
funfun	泛函——非线性数值方法	表A.11
polyfun	多项式和内插函数	表A.12
graphics	通用图形函数	表A.13
plotxy	二维图形函数	表A.14
plotxyz	三维图形函数	表A.15

续表

目　录　名	命　令　函　数	索　　引
lang	语言结构和调试	表 A.16
color	颜色控制和亮度模型函数	表 A.17
strfun	字符串函数	表 A.18
sounds	音频处理函数	表 A.19
iofun	低级文件 I/O 函数	表 A.20
demos	演示例子	·

表 A.2　通 用 命 令

命令	说明
■ 管理命令和函数	
help	在线帮助文本
doc	装入超文本说明
what	M、MAT、MEX 文件的目录列表
type	列出 M 文件
lookfor	通过 help 条目搜索关键字
which	定位函数和文件
demo	运行演示程序
path	控制 MATLAB 的搜索路径
■ 管理变量和工作空间	
who	列出当前变量
whos	列出当前变量(长表)
load	从磁盘文件中恢复变量
save	保存工作空间变量
clear	从内存中清除变量和函数
pack	整理工作空间内存
size	矩阵的尺寸
length	向量的长度
disp	显示矩阵或文本
■ 与文件和操作系统有关的命令	
cd	改变当前工作目录
dir	目录列表
delete	删除文件

续表

■ 与文件和操作系统有关的命令		
	getenv	获取环境变量值
	!	执行操作系统命令
	unix	执行操作系统命令并返回结果
	diary	保存 MATLAB 任务
■ 控制命令窗口		
	cedit	设置命令行编辑
	clc	清命令窗口
	home	光标置左上角
	format	设置输出格式
	echo	MATLAB 文件内使用的回显命令
	more	在命令窗口中控制分页输出
■ 启动和退出 MATLAB		
	quit	退出 MATLAB
	startup	引用 MATLAB 时所执行的 M 文件
	matlabrc	主启动 M 文件
■ 一般信息		
	info	MATLAB 系统信息及 Mathworks 公司信息
	subscribe	成为 MATLAB 的订购用户
	hostid	MATLAB 主服务程序的识别代号
	whatsnew	在说明书中未包含的新信息
	ver	版本信息

表 A.3　操作符和特殊字符

■ 操作符和特殊字符		
	+	加
	-	减
	*	矩阵乘法
	.*	数组乘法
	^	矩阵幂
	.^	数组幂
	\	左除或反斜杠

续表

符号/命令	说明
■ 操作符和特殊字符	
/	右除或斜杠
./	数组除
kron	Kronecker 张量积
:	冒号
( )	圆括号
[ ]	方括号
.	小数点
..	父目录
...	继续
,	逗号
;	分号
%	注释
!	感叹号
'	转置或引用
=	赋值
= =	相等
< >	关系操作符
&	逻辑与
	逻辑或
~	逻辑非
xor	逻辑异或
■ 逻辑函数	
exist	检查变量或函数是否存在
any	向量的任一元为真，则其值为真
all	向量的所有元为真，则其值为真
find	找出非零元素的索引号
isnan	当含 NaN 时，其值为真
isinf	当含无限大元时，其值为真
finite	当含有限值元时，其值为真
isempt	当矩阵为空矩阵时，其值为真
isreal	当矩阵为实矩阵时，其值为真
issparse	当矩阵为稀疏矩阵时，其值为真
isstr	当矩阵为文本串时，其值为真
isglobal	当变量为全局变量时，其值为真

表 A.4　基本数学函数

■ 三角函数		
	sin	正弦
	sinh	双曲正弦
	asin	反正弦
	asinh	反双曲正弦
	cos	余弦
	cosh	双曲余弦
	acos	反余弦
	acosh	反双曲余弦
	tan	正切
	tanh	双曲正切
	atan	反正切
	atan2	四象限反正切
	atanh	反双曲正切
	sec	正割
	sech	双曲正割
	asec	反正割
	asech	反双曲正割
	csc	余割
	csch	双曲余割
	acsc	反余割
	acsch	反双曲余割
	cot	余切
	coth	双曲余切
	acot	反余切
	acoth	反双曲余切
■ 指数函数		
	exp	指数
	log	自然对数
	log10	常用对数
	sqrt	平方根

续表

■ 复数函数		
	abs	绝对值
	argle	相角
	conj	复共轭
	image	复数虚部
	real	复数实部
■ 数值函数		
	fix	朝零方向取整
	floor	朝负无穷大方向取整
	ceil	朝正无穷大方向取整
	round	朝最近的整数取整
	rem	除后余数
	sign	符号函数

表 A.5　特殊数学函数

	besselj	第一类 Bessel(贝塞尔)函数
	bessely	第二类 Bessel 函数
	besseli	改进的第一类 Bessel 函数
	besselk	改进的第二类 Bessel 函数
	beta	β 类函数
	betainc	非完全的 β 函数
	betaln	β 函数的对数
	ellipj	雅可比椭圆函数
	ellipke	完全椭圆积分
	erf	误差函数
	erfc	互补误差函数
	erfcx	比例互补误差函数
	erfinv	逆误差函数
	expint	指数积分函数
	gamma	γ 函数
	gcd	最大公约数
	gammainc	非完全 γ 函数

续表

	lcm	最小公倍数
	log2	分割浮点数
	pow2	比例浮点数
	rat	有理逼近
	rats	有理输出
	cart2pol	变卡笛尔坐标为极坐标
	cart2sph	变卡笛尔坐标为球坐标
	pol2cart	变极坐标为卡笛尔坐标
	sph2cart	变球坐标为卡笛尔坐标

表 A.6　基本矩阵和矩阵操作

■ 基本矩阵		
	zeros	零矩阵
	ones	全“1”矩阵
	eye	单位矩阵
	rand	均匀分布的随机数矩阵
	randn	正态分布的随机数矩阵
	linspace	线性间隔的向量
	logspace	对数间隔的向量
	meshgrid	三维图形的 X 和 Y 数组
	:	规则间隔的向量
■ 特殊变量和常数		
	ans	当前的答案
	eps	相对浮点精度
	realmax	最大浮点数
	realmin	最小浮点数
	pi	圆周率值 3.141 592 653 589 7……
	i, j	虚数单位
	inf	无穷大
	nan	非数值
	flops	浮点运算次数
	nargin	函数输入变量数

续表

■ 特殊变量和常数		
	nargout	函数输出变量数
	computer	计算机类型
	isieee	当计算机采用 IEEE 算术标准时，其值为真
	why	简明的答案
	version	MATLAB 版本号
■ 时间和日期		
	clock	墙上挂钟
	cputime	CPU 时间(以秒为单位)
	date	日历
	etime	计时函数
	tic	秒表开始执行
	toc	秒表停止
■ 矩阵操作		
	diag	建立或提取对角阵
	fliplr	矩阵作左右翻转
	flipud	矩阵作上下翻转
	reshape	改变矩阵大小
	rot90	矩阵旋转 90°
	tril	提取矩阵的下三角部分
	triu	提取矩阵的上三角部分
	:	矩阵的索引号，重新排列矩阵

表 A.7　特 殊 矩 阵

	compan	友矩阵
	gallery	几个小的测试矩阵
	hadamard	Hadamard 矩阵
	hankel	Hankel 矩阵
	hilb	Hilbert 矩阵
	invhilb	逆 Hilbert 矩阵
	kron	Kronecker 张量积
	magic	魔方矩阵
	pascal	Pascal 矩阵

续表

	rosser	经典的对称特征值测试问题
	toeplitz	Toeplitz 矩阵
	vander	Vandermonde 矩阵
	wilkinson	Wilkinson 特征值测试矩阵

表 A.8 矩阵函数——数值线性代数

■ **矩阵分析**		
	cond	计算矩阵条件数
	norm	计算矩阵或向量范数
	rcond	Linpack 逆条件值估计
	rank	计算矩阵秩
	det	计算矩阵行列式值
	trace	计算矩阵的迹
	null	零矩阵
	orth	正交化
	rref	减缩行格式矩阵
■ **线性方程**		
	\ 和 /	线性方程求解
	chol	Cholesky 分解
	lu	高斯消元法求系数阵
	inv	矩阵求逆
	qr	正交三角矩阵分解(简称 QR 分解)
	qrdelete	从 QR 分解中消去一列
	qrinsert	在 QR 分解中插入一列
	nnls	非负最小二乘
	pinv	矩阵伪逆
	lscov	协方差已知的情况下最小二乘求解
■ **特征值和奇异值**		
	eig	求特征值和特征向量
	poly	求特征多项式
	polyeig	多项式特征值问题
	hess	Hessberg 形式

续表

■ 特征值和奇异值		
	qz	广义特征值
	rsf2csf	变实分块对角阵为复对角形式
	cdf2rdf	变复对角矩阵为实分块对角形式
	schur	Schur 分解
	balance	矩阵均衡处理以提高特征值精度
	svd	奇异值分解
■ 矩阵函数		
	expm	矩阵指数
	expm1	实现 expm 的 M 文件
	expm2	通过泰勒级数求矩阵指数
	expm3	通过特征值和特征向量求矩阵指数
	logm	矩阵对数
	sqrtm	矩阵开平方根
	funm	一般矩阵的计算

表 A.9　稀疏矩阵函数

■ 基本稀疏矩阵		
	speye	稀疏单位矩阵
	sprandn	稀疏随机矩阵
	sprandsym	对称的稀疏随机矩阵
	spdiags	从对角阵中形成稀疏矩阵
■ 完全矩阵和稀疏矩阵之间变换		
	sparse	从非零元素及其序号中形成稀疏矩阵
	full	变稀疏矩阵为完全矩阵
	find	找出非零元素的序号
	spconvert	稀疏矩阵外部结构的变换
■ 稀疏矩阵非零元素的处理		
	nnz	非零元素的数目
	nonzeros	非零元素
	nzmax	分配给非零元素的存储量
	spones	用“1”取代非零元素
	spalloc	为非零元素分配内存

续表

■ 稀疏矩阵非零元素的处理		
	issparse	当矩阵为稀疏矩阵时，其值为真
	spfun	只对非零元素取函数
■ 显示稀疏矩阵		
	spy	显示稀疏结构
	gplot	绘图
■ 排序算法		
	colmmd	列最小度
	symmmd	最小对称度
	symrcm	逆 Cathill－Mckee 序
	colperm	基于非零元素按列排序
	randperm	随机排列向量
	dmperm	Dulmage－Mendelsohn 分解
■ 范数、条件数和秩		
	normest	2 范数估计
	condest	1 范数条件估计
	sprank	结构化秩
■ 树型操作		
	treelayout	显示一个或多个结构树
	treeplot	画结构树
	etree	求矩阵的消元树
	etreeplot	画消元树图
■ 其它		
	symbfact	符号分解分析
	spparms	为稀疏矩阵处理过程设置参数
	spaugment	形成最小二乘增广系统

表 A.10　数据分析和傅里叶变换函数

■ 基本操作		
	max	取最大分量
	min	取最小分量
	mean	求均值
	median	求中值

续表

■ 基本操作		
	std	求标准差
	sort	按升序排列
	sum	求各元素之和
	prod	求各元素之积
	cumsum	求元素累积和
	cumprod	求元素累积积
	trapz	利用梯形法计算数值积分
■ 有限差分		
	diff	计算差分和近似微分
	gradient	计算近似梯度
	del2	5 点离散拉普拉斯变换
■ 向量操作		
	cross	向量的矢量积
	dot	向量的点积
■ 相关		
	corrcoef	求相关系数
	cov	求协方差矩阵
	subspace	子空间之间的夹角
■ 滤波和卷积		
	filter	一维数字滤波器
	filter2	二维数字滤波器
	conv	卷积和多项式乘法
	conv2	二维卷积
	deconv	反卷积和多项式除法
■ 傅里叶变换		
	fft	离散傅里叶变换
	fft2	二维离散傅里叶变换
	ifft	离散逆傅里叶变换
	ifft2	二维离散逆傅里叶变换
	abs	取模(绝对值)
	angle	取相角
	unwrap	删除跨越 360°边界的相角
	fftshift	将零点平移到频谱中心

续表

■ 傅里叶变换		
	cplxpair	将数值分类成复共轭对
	nextpow2	最靠近 2 的幂次

表 A.11　泛函——非线性数值方法

	ode23	低阶法求解常微分方程
	ode23p	低阶法求解常微分方程并绘出结果图形
	ode45	高阶法求解常微分方程
	quad	低阶法计算数值积分
	quad8	高阶法计算数值积分
	fmin	单变量函数的极小化
	fmins	多变量函数的极小化
	fzero	找出单变量函数的零点
	fplot	函数绘图

表 A.12　多项式和内插函数

■ 多项式		
	roots	求多项式根
	poly	构造具有指定根的多项式
	polyval	多项式计算
	polyvalm	带矩阵变量的多项式计算
	residue	部分分式展开(留数计算)
	polyfit	数据的多项式拟合
	polyder	微分多项式
	conv	多项式乘法
	deconv	多项式除法
■ 数据内插		
	interp1	一维数据内插(一维查表)
	interp2	二维数据内插(二维查表)
	interpft	利用 FFT 进行一维数据内插
	griddata	数据网格

续表

■ 样条内插		
	spline	3 次样条数据内插
	ppval	分段多项式计算

表 A.13 通用图形函数

■ 建立和控制图形窗口		
	figure	建立图形(图形窗口)
	gcf	获取当前图形的句柄
	clf	消除当前图形
	close	关闭图形
■ 建立和控制坐标系		
	subplot	在标定位置上建立坐标系
	axes	在任意位置上建立坐标系
	gca	获取当前坐标系的句柄
	cla	消除当前坐标系
	axis	控制坐标系的刻度和形式
	caxis	控制伪彩色坐标刻度
	hold	保持当前图形
■ 句柄图形对象		
	figure	建立图形窗口
	axes	建立坐标系
	line	建立曲线
	text	建立文本串
	patch	建立图形填充块
	surface	建立曲面
	image	建立图像
	uicontrol	建立用户界面控制
	uimenu	建立用户界面菜单
■ 句柄图形操作		
	set	设置对象特性
	get	获取对象特性
	reset	重置对象特性
	delete	删除对象

续表

■ 句柄图形操作		
	gco	获取当前对象的句柄
	drawnow	填充未完成绘图事件
	newplot	预测 NextPlot 性质的 M 文件
	findobj	寻找指定特性值的对象
■ 打印和存储		
	print	打印图形或保存图形
	printopt	配置本地打印机缺省值
	orient	设置纸张取向
	capture	屏幕抓取当前图形
■ 动画		
	moviein	初始化动画帧内存
	getframe	获取动画帧
	movie	播放所记录的动画帧
储 其它		
	ginput	用鼠标输入图形
	ishold	返回 Hold 状态
	graymon	设置灰度显示器的图形缺省值
	rbbox	涂抹块
	rotate	沿指定方向旋转对象
	terminal	设置图形终端类型
	uiputfile	弹出保存文件的对话框
	uigetfile	弹出询问文件名的对话框
	whitebg	设置白色背景的图形窗口缺省值
	zoom	二维图形的放大、缩小
	waitforbuttonpress	在图形中等待按键/按钮
■ 用户界面工具		
	dialog	主对话框建立 M 文件
	figflag	当图形为当前显示时其值为真
	layout	定义对话框布局参数
	uiguide	关于用户界面约定/标准/建议的说明
■ 对话框		
	errordlg	建立出错对话框
	helpdlg	建立帮助对话框

续表

■ 对话框		
	questdlg	建立提问对话框
	warndlg	建立警告对话框
■ 打印实用工具		
	prtps	PostScript 打印机驱动程序
	prtwin	MS Windows 驱动程序

表 A.14　二维图形函数

■ 基本 X-Y 图形		
	plot	线性图形
	loglog	对数坐标图形
	semilogx	半对数坐标图形(X 轴为对数坐标)
	semilogy	半对数坐标图形(Y 轴为对数坐标)
	fill	绘制二维多边形填充图
■ 特殊 X-Y 图形		
	polar	极坐标图
	bar	条形图
	stem	离散序列图或杆图
	stairs	阶梯图
	errorbar	误差条图
	hist	直方图
	rose	角度直方图
	compass	区域图
	feather	箭头图
	fplot	绘图函数
	comet	星点图
■ 图形注释		
	title	图形标题
	xlabel	X 轴标记
	ylabel	Y 轴标记
	text	文本注释
	gtext	用鼠标放置文本
	grid	网格线

表 A.15 三维图形函数

■ 曲线和区域填充命令		
	plot3	在三维空间中绘制曲线和点
	fill3	在三维空间中绘制并填充三维多边形
	comet3	三维星点图
■ 三维数据的等高线和其它二维图形		
	contour	等高线图
	contour3	三维等高线图
	clabel	在等高线图上标注高度
	contourc	等高线图计算
	pcolor	伪彩色图
	quiver	箭头图
■ 曲面和网格图形		
	mesh	三维网格曲面
	meshc	网格和等高线混合图形
	meshz	带零平面的三维网格图
	surf	三维曲面阴影图
	surfc	曲面和等高线混合图形
	surfl	带亮度的三维曲面阴影图
	waterfall	落差图
■ 立体可视化		
	slice	立体可视图
■ 图形外形		
	view	指定三维图形视点
	viewmtx	显示变换矩阵
	hidden	设置网格消隐方式
	shading	彩色阴影方式
	axis	坐标轴刻度和外形
	caxis	伪彩色坐标轴刻度
	colormap	颜色对照表
■ 图形注释		
	title	图形标题

续表

■ 图形注释		
	xlable	X 轴标记
	ylable	Y 轴标记
	zlable	Z 轴标记
	text	文本注释
	gtext	用鼠标放置文本
	grid	网格线
■ 三维对象		
	cylinder	产生圆柱体
	sphere	产生球

表 A.16　语言结构和调试

■ MATLAB 编程语言		
	script	有关 MATLAB 的底稿文件和 M 文件的说明
	function	增加新的函数
	eval	执行由 MATLAB 表达式构成的字串
	feval	执行由字串指定的函数
	global	定义全局变量
	nargchk	有效的输入变量数
	lasterr	保持出错信息
■ 程序控制流		
	if	条件执行语句
	else	与 if 命令配合使用
	elseif	与 if 命令配合使用
	end	for，while 和 if 语句的结束
	for	重复执行指定次数(循环)
	while	重复执行不定次数(循环)
	break	终止循环的执行
	return	返回引用的函数
	error	显示信息并终止函数执行
■ 交互输入		
	input	提示用户输入
	keyboard	像底稿文件一样使用键盘输入

续表

■ 交互输入		
	menu	产生由用户输入选择的菜单
	pause	等待用户响应
	uimenu	建立用户界面菜单
	uicontrol	建立用户界面控制
■ 调试命令		
	dbstop	设置断点
	dbclear	删除断点
	dbcont	继续执行
	dbdown	改变局部工作空间的内容
	dbstack	列出调用者
	dbstatus	列出所有断点
	dbstep	执行一条或多条语句
	dbtype	带行号列出 M 文件
	dbup	改变局部工作空间的内容
	dbquit	退出调试状态
	mexdebug	调试 MEX 文件

表 A.17　颜色控制和亮度模型函数

■ 颜色控制		
	colormap	颜色对照表
	caxis	伪彩色坐标轴刻度
	shading	彩色阴影方式
■ 颜色板		
	hsv	色彩——饱和度颜色板
	gray	线性灰度颜色板
	hot	黑—红—黄—白颜色板
	cool	深蓝深红阴影颜色板
	bone	以蓝色为基调的灰度颜色板
	copper	线性青铜色调的颜色板
	pink	线性粉红阴影颜色板
	prism	光谱颜色板
	jet	hsv 颜色板的变型

续表

■ 颜色板		
	flag	红、白、蓝、黑交替的颜色板
■ 颜色板相关函数		
	colorbar	显示颜色条
	hsv2rgb	变 HSV 为 RGB 3 色
	rgb2hsv	变 RGB 为 HSV
	contrast	变灰度颜色板为增强图像对比
	brighten	颜色板加亮或变暗
	spinmap	颜色板旋转
	rgbplot	颜色板绘图
■ 加亮模式		
	surfl	带亮度的三维曲面阴影图
	specular	镜面反射
	diffuse	漫反射
	surfnorm	曲面法线

表 A.18　字符串函数

■ 一般函数		
	strings	MATLAB 中有关字符串函数的说明
	abs	变字符串为数值
	setstr	变数值为字符串
	isstr	当变量为字符串时其值为真
	blanks	空串
	deblank	删除尾部的空串
	str2mat	从各个字符串中形成文本矩阵
	eval	执行由 MATLAB 表达式组成的串
■ 字符串比较		
	strcmp	比较字符串
	findstr	在一字符串中查找另一子串
	upper	变字符串为大写
	lower	变字符串为小写
	isletter	当变量为字母时，其值为真
	isspace	当变量为空白字符时，其值为真

续表

■ 字符串比较		
	strrep	取代字符串
	strtok	在字符串中查找标记
■ 字符串与数值之间变换		
	num2str	变数值为字符串
	int2str	变整数为字符串
	str2num	变字符串为数值
	sprintf	变数值为格式控制下的字符串
	sscanf	变字符串为格式控制下的数值
■ 十进制数与十六进制数之间变换		
	hex2num	变十六进制数为 IEEE 标准下的浮点数
	hex2dec	变十六进制数为十进制数
	dec2hex	变十进制数为十六进制数

表 A.19　音频处理函数

■ 一般音频函数		
	sound	变向量为音频信号
	saxis	音频轴刻度
■ 特定计算机音频函数		
	auwrite	写按 Wu - law 编码的音频文件
	auread	读按 Wu - law 编码的音频文件
	wavwrite	写 MS Windows 的.WAV 音频文件
	wavread	读 MS Windows 的.WAV 音频文件
	mu2lin	变 Wu - law 编码音频信号为线性音频信号
	lin2mu	变线性音频信号为 Wu - law 编码音频信号

表 A.20　低级文件 I/O 函数

■ 打开和关闭文件		
	fopen	打开文件
	fclose	关闭文件
■ 未格式化 I/O		
	fread	从文件读二进制数据

续表

■ 未格式化 I/O		
	fwrite	二进制数据写入到文件
■ 格式化 I/O		
	fscanf	从文件中读格式化的数据
	fprintf	将格式化数据写入到文件
	fgetl	从文件中读行，并丢弃换行符
	fgets	从文件中读行，并保持换行符
■ 文件定位		
	ferror	查询文件 I/O 出错状态
	feof	测试文件尾
	fseek	设置文件位置指针
	ftell	获取文件位置指针
	frewind	反绕文件
■ 字符串变换		
	sprintf	将格式化数据写到字符串
	sscanf	从格式化字符串中读取
■ 文件 I/O		
	wk1const	WK1 记录定义
	wk1read	读 WK1 文件
	wk1write	在 WK1 格式化文件中写矩阵
	wk1wrec	写 WK1 记录头
	csvread	从逗号间隔的格式化文件中读一矩阵
	csvwrite	写一矩阵到逗号间隔的格式化文件中
	dlmread	从以 ASCII 码限界的文件中读一矩阵
	dlmwrite	按 ASCII 码限界的文件格式写一矩阵

附录 B　Toolbox 函数

MATLAB 系统提供了许多 Toolbox（工具箱），而且由于 MATLAB 的可扩充性，Toolbox 的数目与日俱增，这里仅列出一些基本的工具箱函数，以备用户查阅。

表 B.1 为本附录所列出的 Toolbox。表 B.2～B.11 为各个 Toolbox 所提供的工具函数。

表 B.1　MATLAB 提供的部分 Toolbox

目 录 名	工 具 箱 名 称	索　　引
local	局部函数库	表 B.2
signal	信号处理	表 B.3
image	图像处理	表 B.4
control	控制系统	表 B.5
ncd	非线性控制设计	表 B.6
robust	鲁棒控制	表 B.7
ident	系统辨识	表 B.8
optim	最优化	表 B.9
nnet	神经网络	表 B.10
fuzzy	模糊系统	表 B.11

表 B.2　局部函数库

	matlabrc	MATLAB 的主启动 M 文件
	printopt	设置打印选项

表 B.3　信号处理工具箱

■ 波形产生		
	sawtooth	产生锯齿波或三角波
	square	产生方波
	sinc	产生 sinc 或 $\frac{\sin(\pi t)}{\pi t}$ 函数
	diric	产生 Dirichlet 或周期 sinc 函数

续表

■ 滤波器分析和实现		
	abs	取绝对值(幅值)
	angle	取相角
	conv	求卷积
	fftfilt	重叠相加法 FFT 滤波器实现
	filter	直接滤波器实现
	filtfilt	零相位数字滤波
	filtic	filter 函数初始条件选择
	freqs	模拟滤波器频率响应
	freqspace	频率响应中的频率间隔
	freqz	数字滤波器频率响应
	grpdelay	平均滤波延迟(群延迟)
	impz	数字滤波器的冲激响应
	zplane	离散系统零极点图
■ 线性系统变换		
	convmtx	卷积矩阵
	poly2rc	从多项式系数中计算反射系数
	rc2poly	从反射系数中计算多项式系数
	residuez	Z 变换部分分式展开或留数计算
	sos2ss	变系统二阶分割形式为状态空间形式
	sos2tf	变系统二阶分割形式为传递函数形式
	sos2zp	变系统二阶分割形式为零极点增益形式
	ss2sos	变系统状态空间形式为二阶分割形式
	ss2tf	变系统状态空间形式为传递函数形式
	ss2zp	变系统状态空间形式为零极点增益形式
	tf2ss	变系统传递函数形式为状态空间形式
	tf2zp	变系统传递函数形式为零极点增益形式
	zp2sos	变系统零极点增益形式为二阶分割形式
	zp2ss	变系统零极点增益形式为状态空间形式
	zp2tf	变系统零极点增益形式为传递函数形式

续表

■ IIR 滤波器设计		
	besself	Bessel(贝塞尔)模拟滤波器设计
	butter	Butterworth(比特沃思)滤波器设计
	cheby1	Chebyshev(切比雪夫)Ⅰ型滤波器设计
	cheby2	Chebyshev(切比雪夫)Ⅱ型滤波器设计
	ellip	椭圆滤波器设计
	yulewalk	递归数字滤波器设计
■ IIR 滤波器阶的选择		
	buttord	Butterworth 滤波器阶的选择
	cheb1ord	Chebyshev Ⅰ型滤波器阶的选择
	cheb2ord	Chebyshev Ⅱ型滤波器阶的选择
	ellipord	椭圆滤波器阶的选择
■ FIR 滤波器设计		
	fir1	基于窗函数的 FIR 滤波器设计——标准响应
	fir2	基于窗函数的 FIR 滤波器设计——任意响应
	firls	最小二乘 FIR 滤波器设计
	intfilt	内插 FIR 滤波器设计
	remez	Parks-McCellan 最优 FIR 滤波器设计
	remezord	Parks-McCellan 最优 FIR 滤波器阶估计
■ 变换		
	czt	线性调频 Z 变换
	dct	离散余弦变换(DCT)
	idct	逆离散余弦变换
	dftmtx	离散傅里叶变换矩阵
	fft	一维快速傅里叶变换
	ifft	一维逆快速傅里叶变换
	fftshift	重新排列 FFT 的输出
	hilbert	Hilbert(希尔伯特)变换
■ 统计信号处理		
	cov	协方差矩阵
	xcov	互协方差函数估计
	corrcoef	相关系数矩阵

续表

■ **统计信号处理**		
	xcorr	互相关函数估计
	cohere	相关函数平方幅值估计
	csd	互谱密度(CSD)估计
	psd	信号功率谱密度(PSD)估计
	tfe	从输入输出中估计传递函数
■ **窗函数**		
	boxcar	矩形窗
	triang	三角窗
	bartlett	Bartlett(巴特利特)窗
	hamming	Hamming(哈明)窗
	hanning	Hanning(汉宁)窗
	blackman	Blackman(布莱克曼)窗
	chebwin	Chebyshev 窗
	kaiser	Kaiser 窗
■ **参数化建模**		
	invfreqs	模拟滤波器拟合频率响应
	invfreqz	离散滤波器拟合频率响应
	prony	利用 Prony 方法的离散滤波器拟合时间响应
	stmcb	利用 Steiglitz - McBride 迭代方法求线性模型
	levinson	Levinson - Durbin 递归算法
	lpc	线性预测系数
■ **特殊操作**		
	rceps	实倒谱和最小相位重构
	cceps	倒谱分析和最小相位重构
	decimate	降低序列的取样速率
	interp	提高取样速率(内插)
	resample	改变取样速率
	medfilt1	一维中值滤波
	deconv	反卷积和多项式除法
	modulate	通讯仿真中的调制
	demod	通讯仿真中的解调

续表

■ 特殊操作		
	vco	电压控制振荡器
	specgram	频谱分析
■ 模拟原型滤波器设计		
	besselap	Bessel 模拟低通滤波器原型
	buttap	Butterworth 模拟低通滤波器原型
	cheb1ap	Chebyshev Ⅰ 型模拟低通滤波器原型
	cheb2ap	Chebyshev Ⅱ 型模拟低通滤波器原型
	ellipap	椭圆模拟低通滤波器原型
■ 频率变换		
	lp2bp	低通到带通模拟滤波器变换
	lp2hp	低通到高通模拟滤波器变换
	lp2bs	低通到带阻模拟滤波器变换
	lp2lp	低通到低通模拟滤波器变换
■ 滤波器离散化		
	bilinear	双线性变换
	impinvar	冲激响应不变法实现模拟到数字的滤波器变换
■ 其它		
	conv2	二维卷积
	cplxpair	将复数归成复共轭对
	detrend	删除线性趋势
	fft2	二维快速傅里叶变换
	ifft2	二维逆快速傅里叶变换
	filter2	二维数字滤波器
	polystab	稳定多项式
	xcorr2	二维互相关

表 B.4 图像处理工具箱

■ 图像输入/输出		
	bmpread	从磁盘中读 BMP(Microsoft Windows 下的位图)文件
	bmpwrite	将一 BMP 文件写入到磁盘
	gifread	从磁盘中读 GIF 文件
	gifwrite	将 GIF 文件写入磁盘
	hdfpeek	在 HDF 文件中列出目标标记/参考对
	hdfread	从 HDF 文件中读取数据
	hdwrite	写数据到 HDF 文件中
	pcxread	从磁盘中读 PCX 文件
	pcxwrite	将 PCX 文件写入磁盘
	tiffread	从磁盘中读 TIFF 文件
	tiffwrite	将 TIFF 文件写入磁盘
	xwdread	从磁盘中读 XWD 文件
	xwdwrite	将 XWD 文件写入磁盘
■ 实用程序		
	getimage	从坐标系中读取图像数据
	isbw	当图像为黑白图像时，其值为真
	isgray	当图像为灰度图像时，其值为真
	isind	当图像为加标图像时，其值为真
■ 颜色操作		
	brighten	加亮或增暗一颜色板
	cmunique	寻找唯一的颜色板及相应的图像
	cmpermute	置换颜色板位置
	cmgamma	γ 校正颜色板
	cmgamdef	缺省的 γ 校正表
	dither	Floyd - Steinberg 图像颤抖算法
	hsv2rgb	变 HSV 值为 RGB 颜色空间
	imadjust	调整并增强图像强度
	imapprox	利用更少颜色的图像逼近加标图像

续表

■ **颜色操作**		
	ntsc2rgb	变 NTSC 值为 RGB 颜色空间
	rgb2gray	变 RGB 值为灰度值
	rgb2hsv	变 RGB 值为 HSV 颜色空间
	rgb2ntsc	变 RGB 值为 NTSC 颜色空间
	rgbplot	绘制 RGB 颜色板分量的图形
■ **几何操作**		
	imcrop	修剪图像
	imresize	改变图像大小
	imrotate	旋转图像
	truesize	改变图像大小使之具有实际尺寸
	imzoom	放大或缩小图像和二维图形
■ **图像增强/分析**		
	brighten	增强或削弱颜色板
	grayslice	密度(强度)限幅
	histeq	直方图均衡化
	imadjust	调整和展宽图像强度
	imapprox	利用较少颜色的图像逼近图像
	imhist	图像直方图
	impixel	-像素点的颜色
	improfile	轮廓强度
	interp2	二维数据内插
■ **图像统计**		
	mean2	矩阵的均值
	corr2	二维相关系数
	std2	二维标准差
■ **形态操作**		
	bwarea	二进制图像中的目标区域
	dilate	加浓二进制图像
	erode	冲淡二进制图像
	edge	边界提取
	bweuler	欧拉数
	bwmorph	形态算子
	bwperim	二进制图像中目标的周围

续表

■ **FIR(有限冲激响应)滤波器设计**		
	fsamp2	通过频率取样的二维 FIR 滤波器设计
	fspecial	特殊的二维滤波器
	ftrans2	通过频率变换的二维 FIR 滤波器设计
	fwind1	使用一维窗函数的 FIR 滤波器设计
	fwind2	使用二维窗函数的 FIR 滤波器设计
	imnoise	图像噪声
■ **频率响应**		
	freqspace	二维频率响应的频率空间
	freqz2	二维频率响应
■ **滤波**		
	colfilt	局部非线性滤波
	conv2	二维卷积
	filter2	二维滤波
	medfilt2	二维中值滤波
	mfilter2	屏蔽滤波
	nlfilter	局部非线性滤波
	wiener2	自适应二维维纳滤波
■ **分块处理**		
	bestblk	分块处理的最佳块大小
	blkproc	按块处理一图像
	col2im	重新排列以形成图像
	colfilt	局部非线性滤波
	im2col	重新排列成列
■ **个别区域**		
	mfilter2	屏蔽滤波
	roipoly	定义感兴趣的多边区域
	roicolor	用颜色定义感兴趣的区域
■ **变换**		
	dct2	二维离散余弦变换
	fft2	二维快速傅里叶变换
	fftshift	零频移到频谱中心
	idct2	二维逆离散余弦变换

续表

■ 变换		
	ifft2	二维逆快速傅里叶变换
	radon	Radon 变换
■ 转换		
	dither	Floyd - Steinberg 图像颤抖
	gray2ind	变灰度图像为附标图像
	hsv2rgb	变 HSV 值为 RGB 值
	im2bw	变图像为黑白图形
	imslice	在图像中获取/置入图像块
	ind2gray	变附标图像为灰度图像
	ind2rgb	变附标图像为 RGB 图像
	mat2gray	变矩阵为(灰度)图像
	ntsc2rgb	变 NTSC 值为 RGB 值
	rgb2gray	变 RGB 图像或值为灰度图像或值
	rgb2hsv	变 RGB 值为 HSV 值
	rgb2ind	变 RGB 图像为附标图像
	rgb2ntsc	变 RGB 值为 NTSC 值
■ 图像显示		
	colorbar	显示颜色条
	colormap	设置或获取颜色查找表
	gray	线性灰度颜色板
	hsv，hot，jet	颜色板
	image	显示附标图像
	imagesc	数据定标并按图像显示
	imcontour	图像等高线
	immovie	制作图像动画
	imshow	显示所有类型的图像数据
	montage	按矩形剪辑方式显示图像
	subimage	显示多个图像
	warp	将图像卷成曲面
■ 演示		
	imdemo	一般图像处理演示
	dctdemo	二维离散余弦变换图像压缩演示
	firdemo	二维 FIR 滤波器演示

续表

■ 演示		
	nlfdemo	二维非线性滤波演示
■ 专用函数		
	cumsum3d	三维矩阵封装成二维矩阵时的累积和
	dct	一维离散余弦变换
	dctmtx2	一元二维离散余弦变换矩阵
	ditherc	图像颤抖的 MEX 文件
	elem3d	三维矩阵封装成二维矩阵的元素位置
	getline	利用橡皮线跟踪鼠标移动
	getpts	利用可视点跟踪鼠标移动
	getrect	利用橡皮矩形跟踪鼠标移动
	gif	压缩 GIF 数据
	hdfreadc	读 HDF 文件的 MEX 文件
	hdfpeekc	搜索 HDF 文件的 MEX 文件
	hdfwc	写 HDF 文件的 MEX 文件
	idct	一维逆离散余弦变换
	im2gray	变图像为灰度
	imhistc	图像直方图计算的 MEX 文件
	ndx3d	三维矩阵封装成二维矩阵的索引
	rgb2im	变 RGB 图像为附标或强度图像
	rle	压缩编码数据
	tiff	压缩 tiff 编码数据
	vmquant	与彩色量化 MEX 文件接口的 M 文件
	waitbar	显示等待条
■ MAT 文件		
	bwmorph. mat	bwmorph. m 文件的查找表
	forest. mat	Carmanah Old Growth Forest 的扫描相片
	mri. mat	人体心脑的磁性共振图像
	trees. mat	树的扫描图像

表 B.5 控制系统工具箱

■ 建模		
	append	追加系统动态特性
	augstate	变量状态作为输出
	blkbuild	从方框图中构造状态空间系统
	cloop	系统的闭环
	connect	方框图建模
	conv	两个多项式的卷积
	destim	从增益矩阵中形成离散状态估计器
	dreg	从增益矩阵中形成离散控制器和估计器
	drmodel	产生随机离散模型
	estim	从增益矩阵中形成连续状态估计器
	feedback	反馈系统连接
	ord2	产生二阶系统的 A、B、C、D
	pade	时延的 Pade 近似
	parallel	并行系统连接
	reg	从增益矩阵中形成连续控制器和估计器
	rmodel	产生随机连续模型
	series	串行系统连接
	ssdelete	从模型中删除输入、输出或状态
	ssselect	从大系统中选择子系统
■ 模型变换		
	c2d	变连续系统为离散系统
	c2dm	利用指定方法变连续为离散系统
	c2dt	带一延时变连续为离散系统
	d2c	变离散为连续系统
	d2cm	利用指定方法变离散为连续系统
	poly	变根值表示为多项式表示
	residue	部分分式展开
	ss2tf	变状态空间表示为传递函数表示
	ss2zp	变状态空间表示为零极点表示
	tf2ss	变传递函数表示为状态空间表示
	tf2zp	变传递函数表示为零极点表示
	zp2tf	变零极点表示为传递函数表示
	zp2ss	变零极点表示为状态空间表示

续表

■ 模型简化		
	balreal	平衡实现
	dbalreal	离散平衡实现
	dmodred	离散模型降阶
	minreal	最小实现和零极点对消
	modred	模型降阶
■ 模型实现		
	canon	正则形式
	ctrbf	可控阶梯形
	obsvf	可观阶梯形
	ss2ss	采用相似变换
■ 模型特性		
	covar	相对于白噪声的连续协方差响应
	ctrb	可控性矩阵
	damp	阻尼系数和固有频率
	dcgain	连续稳态(直流)增益
	dcovar	相对于白噪声的离散协方差响应
	ddamp	离散阻尼系数和固有频率
	ddcgain	离散稳态(直流)增益
	dgram	离散可控性和可观性
	dsort	按幅值排序离散特征值
	eig	特征值和特征向量
	esort	按实部排序连续特征值
	gram	可控性和可观性
	obsv	可观性矩阵
	printsys	按格式显示系统
	roots	多项式之根
	tzero	传递零点
	tzero2	利用随机扰动法传递零点
■ 时域响应		
	dimpulse	离散时间单位冲激响应
	dinitial	离散时间零输入响应
	dlsim	任意输入下的离散时间仿真
	dstep	离散时间阶跃响应

续表

■ 时域响应		
	filter	单输入单输出 Z 变换仿真
	impulse	冲激响应
	initial	连续时间零输入响应
	lsim	任意输入下的连续时间仿真
	ltitr	低级时间响应函数
	step	阶跃响应
	stepfun	阶跃函数
■ 频域响应		
	bode	Bode(波特)图(频域响应)
	dbode	离散 Bode 图
	dnichols	离散 Nichols 图
	dnyquist	离散 Nyquist 图
	dsigma	离散奇异值频域图
	fbode	连续系统的快速 Bode 图
	freqs	拉普拉斯变换频率响应
	freqz	Z 变换频率响应
	ltifr	低级频率响应函数
	margin	增益和相位裕度
	nichols	Nichols 图
	ngrid	画 Nichols 图的栅格线
	nyquist	Nyquist 图
	sigma	奇异值频域图
■ 根轨迹		
	pzmap	零极点图
	rlocfind	交互式地确定根轨迹增益
	rlocus	画根轨迹
	sgrid	在 ω_n, z 网格上画连续根轨迹
	zgrid	在 ω_n, z 网格上画离散根轨迹
■ 增益选择		
	acker	单输入单输出极点配置
	dlqe	离散线性二次估计器设计
	dlqew	离散线性二次估计器设计
	dlqr	离散线性二次调节器设计

续表

■ **增益选择**		
	dlqry	输出加权的离散调节器设计
	lqe	线性二次估计器设计
	lqed	基于连续代价函数的离散估计器设计
	lqe2	利用 Schur 法设计线性二次估计器
	lqew	一般线性二次估计器设计
	lqr	线性二次调节器设计
	lqrd	基于连续代价函数的离散调节器设计
	lqry	输出加权的调节器设计
	lqr2	利用 Schur 法设计线性二次调节器
	place	极点配置
■ **方程求解**		
	are	代数 Riccati 方程求解
	dlyap	离散 Lyapunov 方程求解
	lyap	连续 Lyapunov 方程求解
	lyap2	利用对角化求解 Lyapunov 方程
■ **演示示例**		
	ctrldemo	控制工具箱介绍
	boildemo	锅炉系统的 LQG 设计
	jetdemo	喷气式飞机偏航阻尼的典型设计
	diskdemo	硬盘控制器的数字控制
	kalmdemo	Kalman 滤波器设计和仿真
■ **实用工具**		
	abcdchk	检测(A，B，C，D)组的一致性
	chop	取 n 个重要的位置
	dexresp	离散取样响应函数
	dfrqint	离散 Bode 图的自动定范围的算法
	dfrqint2	离散 Nyquist 图的自动定范围的算法
	dmulresp	离散多变量响应函数
	dists1	到直线间的距离
	dric	离散 Riccati 方程留数计算
	dsigma2	DSIGMA 实用工具函数
	dtimvec	离散时间响应的自动定范围算法
	exresp	取样响应函数

续表

■ 实用工具		
	freqint	Bode 图的自动定范围算法
	freqint2	Nyquist 图的自动定范围算法
	freqresp	低级频率响应函数
	givens	旋转
	housh	构造 Householder 变换
	imargin	利用内插技术求增益和相位裕度
	lab2ser	变标号为字符串
	mulresp	多变量响应函数
	nargchk	检测 M 文件的变量数
	perpxy	寻找最近的正交点
	poly2str	变多项式为字符串
	printmat	带行列号打印矩阵
	ric	Riccati 方程留数计算
	schord	有序 Schwr 分解
	sigma2	SIGMA 实用工具函数
	tfchk	检测传递函数的一致性
	timvec	连续时间响应的自动定范围算法
	tzreduce	在计算过零点时简化系统
	vsort	匹配两根轨迹的向量

表 B.6　非线性控制设计工具箱

■ 对话框管理		
	coneddlg	管理 NCD 工具箱固定编辑器的对话框
	paramdlg	管理 NCD 优化参数的对话框
	rangedlg	管理坐标系范围的对话框
	refdlg	管理 NCD 参考信号的对话框
	stepdlg	管理 NCD 阶跃响应的对话框
	uncerdlg	管理 NCD 不确定变量的对话框
■ 主要界面		
	contrncd	建立 NCD 固定图形的用户界面控制
	menuncd	建立 NCD 固定图形的用户界面菜单
	ncdblock	包含 NCD 框图的 SIMULINK 系统
	optblock	打开一个 NCD 图形的底稿文件
	optfig	建立一个 NCD 固定图形

续表

■ **主要优化**		
	costfun	NCD 优化的代价函数
	nlinopt	执行优化算法
■ **演示示例**		
	ncddemo	包含所有 NCD 演示示例的 SIMULINK 系统
	ncddemo1	PID 控制器
	ncddemo2	带前馈控制器的 LQR
	ncddemo3	多输入多输出的 PI 控制器
	ncddemo4	倒摆演示
■ **教程**		
	ncdtut1	控制设计示例
	ncdtut2	系统辨识示例
■ **用户界面工具**		
	dialog	主对话框建立 M 文件
	errordlg	建立出错对话框
	figflag	当图形为当前显示在屏幕上时，其值为真
	helpdlg	显示一帮助对话框
	layout	定义对话框布局参数的底稿文件
	questdlg	建立提问对话框
	uiguide	有关用户界面约定/标准/建议的说明
	warndlg	建立警告对话框
■ **演示和教程实用工具**		
	ncd1init	为 ncddemo1 的优化进行设置
	ncd2init	为 ncddemo2 的优化进行设置
	ncd3init	为 ncddemo3 的优化进行设置
	ncd4init	为 ncddemo4 的优化进行设置
	penddata	为 ncdtut2(即倒摆)进行设置
■ **界面实用工具**		
	curobj	提供有关当前点的信息
	dividecb	将固定界分为两部分
	delline	从 NCD 图中删除所有的图
	donep	收回 Close 按钮和菜单
	errorncd	管理 NCD 产生的常见错误，它调用 errordlg(出错对话框)

续表

■ 界面实用工具		
	fillaxes	建立约束边界并进行数据检测
	forceit	在已存在的界限内插入一子集
	keyncd	NCD 按键函数
	loadncd	装入并显示 NCD 数据
	makesurf	建立并限界曲面
	snapncd	以 22.5°间隔排出约束条
	refresho	使约束矩阵与图形一致
	saveload	当文件是从 SelectFile 中选择时，其值为真
	texted	收回 Port 可编辑的文本
	undoncd	放弃上次 NCD 图形用户界面的操作
	updatdlg	更新 NCD 对话框
■ 最优化实用工具		
	convertm	变约束矩阵为最优化格式
	minipars	NCD 最小化分析
	montevar	初始化 Monte Carlo 仿真
	ncdglob	定义 NCD 全局变量
	str2mat2	变一行字符串为多行字符串
■ 帮助文本文件(以.HLP 为扩展名)		
	hotkey	热键帮助
	mainncd	一般 NCD 帮助
	paramdlg	最优化参数对话框的帮助
	readncd	与 README.M 文件内容相同
	stepdlg	阶跃响应对话框的帮助
	uncerdlg	不确定性变量对话框的帮助

表 B.7　鲁棒控制工具箱

■ 可选系统数据结构		
	branch	从树中提取一分支
	graft	在树中增加一分支
	issystem	辨识一系统变量
	istree	辨识一树型变量

续表

■ **可选系统数据结构**		
	mksys	为系统建立树变量
	tree	建立树变量
	vrsys	返回标准系统变量名
■ **建模**		
	augss	系统增广(状态空间模型)
	augtf	系统增广(传递函数模型)
	interc	一般多变量内连系统
建 模型转换		
	bilin	多变量双线性变换
	des2ss	利用奇异值分解变系统为状态空间系统
	lftf	线性分式变换
	sectf	扇形变换
	stabproj	稳定和逆稳定映射
	slowfast	慢/快分解
	tfm2ss	变传递函数模型为状态空间模型
■ **实用工具**		
	aresolv	广义连续时间 Riccati 方程求解
	daresolv	广义离散时间 Riccati 方程求解
	riccond	连续时间 Riccati 方程的条件数
	driccond	离散时间 Riccati 方程的条件数
	blkrsch	通过 cschur 得到块有序实 Schur 形式
	cschur	通过复旋转得有序复 Schur 形式
■ **多变量 Bode 图**		
	cgloci	连续特性增益轨迹
	dcgloci	离散特性增益轨迹
	dsigma	离散奇异值 Bode 图
	muopt	具有实/复数混合不确定性系统的 SSV(结构化奇异值)上界
	osborne	通过 Osborne 法求得的 SSV 上界
	perron	计算 Perron 特征值
	psv	Perron 特征结构的 SSV
	sigma	连续奇异值 Bode 图
	ssv	结构化奇异值 Bode 图

续表

■ **因子分解技术**		
	iofc	内外因子分解(列类型)
	iofr	内外因子分解(行类型)
	sfl	左边频谱分解
	sfr	右边频谱分解
■ **模型简化方法**		
	balmr	截断均衡模型简化
	bstschml	相对误差 Schur 模型简化
	bstschmr	相对误差 Schur 模型简化
	imp2ss	从脉冲响应到状态空间实现
	obalreal	有序均衡实现
	ohklmr	最优 Hankel 极小化逼近
	rschur	Schur 模型简化
■ **鲁棒控制综合方法**		
	h2lqg	连续时间 H_2 综合
	dh2lqg	离散时间 H_2 综合
	hinf	连续时间 H_∞ 综合
	dhinf	离散时间 H_∞ 综合
	hinfopt	H_∞ 综合的 γ 迭代
	normh2	计算 H_2 范数
	normhinf	计算 H_∞ 范数
	lqg	LQG 最优控制综合
	ltru	LQG 闭环传递补偿
	ltry	LQG 闭环传递补偿
	youla	Youla 参数化
■ **演示示例**		
	accdemo	弹簧质量标准问题
	dintdemo	双积分器系统的 H_∞ 设计
	hinfdemo	飞机或大型空间结构的 H_2 或 H_∞ 设计示例
	ltrdemo	LQR/LTR 设计示例:飞机
	mudemo	μ 综合示例
	mudemo1	μ 综合示例
	mrdemo	鲁棒模型简化示例
	rctdemo	鲁棒控制工具箱演示——主菜单

表 B.8　系统辨识工具箱

■ 仿真和预测		
	idsim	仿真一给定的系统
	pe	计算预测误差
	poly2th	从给定的多项式中构造 Θ 矩阵
	predict	M 步超前预测
■ 数据处理		
	dtrend	从数据集中删除方位
	idfilt	通过 Butterworth 滤波器对数据进行滤波
■ 非参数化估计		
	covf	估计数据矩阵的协方差矩阵
	cra	相关分析
	etfe	估计经验传递函数并计算周期图
	spa	频谱分析
■ 参数估计		
	ar	利用各种方法的 AR 信号模型
	armax	ARMAX 模型预测误差估计
	arx	ARX 模型的最小二乘估计
	bj	Box - Jenkins 模型的预测误差估计
	canstart	具有初值参数估计的多变量模型
	ivar	时间序列的 AR 部分的仪器 Ⅳ 估计
	ivx	单输出 ARX 模型的仪器可变估计
	iv4	ARX 模型近似最优的 Ⅳ 估计
	oe	输出误差模型的预测误差估计
	pem	一般线性模型的预测误差估计
■ 建立模型结构		
	arx2th	ARX 模型的 Θ 格式
	canform	正则形模型结构
	mf2th	将用户定义的模型结构封装入 Θ 模型格式中
	modstruc	在 ms2th 函数中使用的模型结构
	ms2th	将标准状态空间参数封装入 Θ 格式中
	poly2th	从给定多项式中产生 Θ 矩阵

续表

■ 处理模型结构		
	fixpar	在状态空间和 ARX 模型结构中，找出要修正的参数
	sett	在 Θ 结构中设置取样间隔
	thinit	参数的(随机)初始值
	unfixpar	在状态空间和 ARX 模型结构中，放松参数
■ 模型变换		
	th2arx	变 Θ 格式模型为 ARX 模型
	th2ff	求模型的频率响应及标准偏差
	th2par	变 Θ 格式为参数和协方差阵
	th2poly	求给定模型相应的多项式
	th2ss	变 Θ 格式为状态空间表示
	th2tf	变 Θ 格式为传递函数表示
	th2zp	求零极点、静态增益和标准偏差
	thc2thd	变连续时间模型为离散时间模型
	thd2thc	变离散时间模型为连续时间模型
■ 模型表示		
	bodeplot	传递函数的 Bode 图或频谱
	ffplot	频域函数
	idplot	输入-输出数据
	nyqplot	传递函数的 Nyquist 图
	present	屏幕上的参数模型
	zpplot	零点和极点
■ 信息提取		
	getmfth	获取定义模型结构的 M 文件的文件名
	getncap	获取数据点数和参数个数
	getff	选取频率函数
	gett	为某模型获取取样间隔
	getzp	在由 th2zp 函数产生的零极点格式中，提取零点和极点
■ 模型合法化		
	compare	将仿真和预测的输出与测量输出比较
	idsim	仿真一给定的系统
	pe	预测误差

续表

■ 模型合法化		
	predict	M 步超前预测
	resid	计算和测试与某模型相关的留数
■ 估计模型的不确定性		
	idsimsd	在仿真模型响应中说明不确定性
	th2ff	模型频率函数和标准偏差
	th2zp	零点、极点、静态增益及其标准偏差
■ 模型结构选择		
	arxstruc	ARX 模型类的损失函数
	ivstruc	单输出类的输出误差拟合
	selstruc	根据各种准则选择模型结构
	struc	arxstruc 和 ivstruc 的典型结构矩阵
■ 递归参数估计		
	rarx	对 AR 模型递归计算估值
	rarmax	对 ARMAX 模型递归计算估值
	rbj	对 Box - Jenkins 模型递归计算估值
	roe	对输出误差模型递归计算估值
	rpem	对一般模型递归计算估值
	rplr	对一般模型递归计算估值
	segment	分段数据并跟踪快变系统

表 B.9 最优化工具箱

■ 非线性最小化函数		
	attgoal	达到多目标
	constr	约束极小化
	fmin	无约束极小化(标量情况)
	fminu	利用梯度搜索的无约束极小化
	fmins	利用单纯形搜索的无约束极小化
	fsolve	非线性方程求解
	leastsq	非线性最小二乘
	minimax	极小极大求解
	seminf	半定极小化

续表

■ **矩阵问题极小化**		
	lp	线性规划
	qp	二次规划
	nnls	非负最小二乘
■ **控制缺省值和选项**		
	foptions	参数设置
■ **演示**		
	optdemo	演示菜单
	tutdemo	启动教程
	bandemo	香蕉型函数的极小化
	goaldemo	目标达到
	dfildemo	有限精度滤波器设计
	datdemo	数据拟合成曲线
■ **内部使用的实用程序**		
三次内插程序		
	cubic	内插4点以找出极大值
	cubici1	内插2点和梯度，以估计极小值
	cubici2	内插3点和1梯度
	cubici3	内插2点和梯度，以找出步长和极小值
二次内插程序		
	quad2	内插3点以找出极大值
	quadinter	内插3点以估计极小值
演示实用程序		
	eigfun	返回分类特征值的函数
	elimone	消去一变量
	filtfun	频率响应和根
	filtfun2	频率响应范数和根
	fitfun	返回拟合数据中的误差范数
	fitfun2	返回拟合数据中的误差矢量
半定实用程序		
	semifun	半定问题转换成约束问题
	findmax	在数据向量中内插极大值

续表

半定实用程序		
	findmax2	在数据矩阵中内插极大值
	v2sort	分类两向量，然后删去丢失的元素
目标达到的实用程序		
	goalfun	目标达到问题转换成约束条件问题
	goalgra	变换目标达到问题中的梯度
测试程序		
	toptim	最优化测试组
	toptimf	最优化测试组的测试函数
	toptimg	最优化测试组的测试函数梯度
其它		
	graderr	用于检查梯度的不一致性
	lsint	初始化最小二乘程序的函数
	optint	初始化无约束极小化程序的函数
	searchq	线性搜索程序

表 B.10 神经网络工具箱

■ 误差分析函数		
	errsurf	计算误差曲面
	plotep	在误差曲面上绘制权和基位置图
	plotes	绘制误差曲面图
■ δ 函数		
	deltalin	对 PURELIN 神经元的 δ 函数
	deltalog	对 LOGSIG 神经元的 δ 函数
	deltatan	对 TANSIG 神经元的 δ 函数
■ 设计		
	solvehop	设计 Hopfield 网络
	solvelin	设计线性网络
	solverb	设计径向基网络
	solverbe	设计精确的径向基网络

续表

■ 初始化	
initc	竞争层初始化
initelm	Elman 递归网络初始化
initff	至多三层的前向网络初始化
initlin	线性层初始化
initlvq	LVQ 网络初始化
initp	感知层初始化
initsm	自组织映射初始化
midpoint	产生中点值
nwlog	对 LOGSIG 神经元产生 Nguyen－Widrow 随机数
nwtan	对 TANSIG 神经元产生 Nguyen－widrow 随机数
randnc	产生归一化列随机数
randnr	产生归一化行随机数
rands	产生对称随机数
■ 学习规则	
learnbp	反向演播学习规则
learnbpm	带预测的反向演播学习规则
learnh	Hebb 学习规则
learnhd	退化的 Hebb 学习规则
learnis	内星学习规则
learnk	Kohonen 学习规则
learnlm	Levenberg－Marquardt 学习规则
learnlvq	学习矢量量化规则
learnos	外星学习规则
learnp	感知层学习规则
learnpn	归一化的感知层学习规则
learnwh	Widrow－Hoff 学习规则
■ 矩阵	
combvec	创建所有的矢量集
delaysig	从信号矩阵中建立退化的信号矩阵
dist	计算矢量距离

续表

	函数	说明
■ 矩阵		
	ind2vec	变下标矢量为稀疏矩阵表示
	normc	归一化矩阵列
	normr	归一化矩阵行
	pnormc	伪归一化矩阵列
	quant	离散化成某数值的整数倍
	sumsqr	平方和
	vect2ind	变稀疏矩阵表示为下标矢量
■ 邻域		
	nbdist	使用矢量距离的邻域阵
	nbgrid	使用栅格距离的邻域阵
	nbman	使用 Manhattan 距离的邻域阵
■ 绘图		
	barerr	每个输出矢量的误差条形图表
	hintonw	绘制权值图
	hintonwb	绘制权值和偏差图
	ploterr	绘出网络误差与时间的关系
	plotes	绘制误差曲面
	plotfa	绘出目标模式及网络函数的逼近
	plotpv	绘出限幅神经元的感知器分类
	plotsm	绘制自组织映射图
	plottr	绘出网络误差记录及自适应学习速率
	plotvec	用不同颜色绘制矢量
■ 仿真		
	simuc	竞争层仿真
	simuelm	Elman 递归网络仿真
	simuff	前向网络仿真
	simuhop	Hopfield 网络仿真
	simulin	线性层仿真
	simup	感知层仿真
	simurb	径向基网络仿真
	simusm	自组织映射仿真

续表

■ 训练		
	trainbp	利用反向演播训练前向网络
	trainbpx	利用快速反向演播训练网络
	trainc	训练竞争层网络
	trainelm	训练 Elman 递归网络
	trainlvq	训练 LVQ 网络
	trainp	利用感知规则训练感知层
	trainpn	利用归一化感知规则训练感知层
	trainsm	利用 Kohonen 规则训练自组织映射
	trainwh	利用 Widrow Hoff 规则训练线性层
■ 传递函数		
	compet	竞争层传递函数
	hardlim	硬限幅传递函数
	hardlims	对称硬限幅传递函数
	logsig	对数 S 型传递函数
	purelin	线性传递函数
	radbas	径向基传递函数
	satlins	对称饱和线性传递函数
	tansig	正切 S 型传递函数

表 B.11　模糊系统工具箱

■ GUI 编辑器		
	fuzzy	基本 FIS(模糊推理系统)编辑器
	mfedit	隶属度函数编辑器
	ruleedit	规则编辑器及(句法)分析程序
	ruleview	规则观察器及模糊推理框图
	surfview	输出曲面观测器
■ 隶属度函数		
	dsigmf	两个“S”形隶属度函数的差
	gauss2mf	双边高斯曲线隶属度函数
	gaussmf	高斯曲线隶属度函数
	gbellmf	广义钟形隶属度函数

续表

函数	说明
pimf	π 形隶属度函数
psigmf	两个“S”形隶属度函数的积
smf	“S”形隶属度函数
sigmf	“sigmoid(S)”形隶属度函数
trapmf	梯形隶属度函数
trimf	三角形隶属度函数
zmf	“Z”形隶属度函数
■ 命令行 FIS 函数	
addmf	将隶属度函数加到 FIS 中
addrule	将规则加到 FIS 中
addvar	将变量加到 FIS 中
defuzz	去模糊隶属度函数
evalfis	完成模糊推理计算
evalmf	隶属度函数计算
gensurf	产生 FIS 输出曲面
getfis	获得模糊系统的特性
mf2mf	在函数之间变换参数
newfis	产生新的 FIS
parsrule	分析模糊规则
plotfis	显示 FIS 输入/输出图
plotmf	显示出一个变量的所有隶属度函数
readfis	从磁盘中装入 FIS
rmmf	从 FIS 删除隶属度函数
rmvar	从 FIS 中删除变量
setfis	设置模糊系统特性
showfis	显示带注释的 FIS
showrule	显示 FIS 规则
writefis	在磁盘中保存 FIS
■ 先进技术	
anfis	Sugeno　type FIS 的训练程序
fcm	利用模糊 C 平均聚集方法找出簇
genfis1	利用一般方法产生 FIS 矩阵
genfis2	利用减法聚集方法产生 FIS 矩阵
subclust	利用减法聚集方法估计簇中心

表 B.12　小波分析工具箱

■ **小波分析中的通用函数**		
	biorfilt	双正交小波滤波器组
	dyaddown	二元取样
	dyadup	二元插值
	wavefun	小波函数和尺度函数
	intwave	积分小波函数 ψ
	orthfilt	正交小波滤波器组
	qmf	镜像二次滤波器
	wfilters	小波滤波器
	wavemngr	小波管理
	wmaxlev	计算小波分解的最大尺度
	deblankl	把字符串变成无空格的小写字符串
	errargn	检查函数参数数目
	errargt	检查函数的参数类型
	num2mstr	把数字转化为字符串
	wcodemat	对矩阵进行量化编码
	wcommon	寻找公共元素
	wkeep	提取向量或矩阵中的一部分
	wrev	向量逆序
	nstdfft	非标准一维快速博里叶变换(FFT)
	instdfft	非标准一维快速逆博里叶变换
	std	计算标准差
■ **小波函数**		
	biorwavf	双正交样条小波滤波器
	coifwavf	Coiflets 小波滤波器
	dbaux	Daubechies 小波滤波器
	dbwavf	Daubechies 小波滤波器
	mexihat	墨西哥帽小波
	meyer	meyer 小波
	meyeraux	Meyer 小波辅助函数
	morlet	Morlet 小波
	symwavf	Symlets 小波滤波器

续表

函数	说明
■ 一维小波变换	
cwt	一维连续小波变换
dwt	单尺度一维离散小波变换
dwtmode	离散小波变换拓展模式
idwt	单尺度一维离散小波逆变换
dwtper	单尺度一维离散小波变换(周期性)
idwtper	单尺度一维离散小波重构(周期性)
wavedec	多尺度一维小波分解
appcoef	提取一维小波变换低频系数
detcoef	提取一维小波变换高频系数
waverec	多尺度一维小波重构
upwlev	单尺度一维小波分解的重构
wrcoef	对一维小波系数进行单支重构
upcoef	一维系数的直接小波重构
■ 二维小波变换	
dwt2	单尺度二维离散小波变换
idwt2	单尺度逆二维离散小波变换
dwtper2	单尺度二维离散小波变换(周期性)
idwtper2	单尺度二维离散逆小波分析(周期性)
wavedec2	多尺度二维小波分解
waverec2	多尺度二维小波重构
appcoef2	提取二维小波分解低频系数
detcoef2	提取二维小波分解高频系数
upwlev2	二维小波分解的单尺度重构
wrcoef2	对二维小波系数进行单支重构
upcoef2	二维小波分解的直接重构
■ 小波包算法	
wpdec	一维小波包的分解
wprec	一维小波包分解的重构
wpdec2	二维小波包的分解
wprec2	二维小波包分解的重构
wpcoef	计算小波包系数
wprcoef	小波包分解系数的重构
wpfun	小波包函数
wpsplt	分割(分解)小波包
wpjoin	重新组合小波包
wpcutree	剪切小波包分解树

续表

■ 小波包算法		
	besttree	计算最佳(优)树
	bestlevt	计算完整最佳小波包树
	wp2wtree	从小波包树中提取小波树
	wentropy	计算小波包的熵
	entrupd	更新小波包的熵值
■ 信号和图像的消噪与压缩		
	ddencmp	获取默认值阈值(软或硬)、熵标准
	thselect	信号消噪的阈值选择
	wden	用小波进行一维信号的自动消噪
	wdencmp	用小波进行信号的消噪或压缩
	wnoise	产生含噪声的测试函数数据
	wnoisest	估计一维小波的系数的标准偏差
	wpdencmp	用小波包进行信号的消噪或压缩
	wpthcoef	进行小波包分解系数的阈值处理
	wthcoef	一维信号的小波系数阈值处理
	wthcoef2	二维信号的小波系数阈值处理
	wthresh	进行软阈值或硬阈值处理
■ 树操作应用函数		
	maketree	创建小波(包)分解树
	plottree	画树结构图形
	nodejoin	重组结点
	allnodes	计算树结点
	depo2ind	将深度—位置结点形式转化成索引结点形式
	ind2depo	将索引结点形式转化成深度—位置结点形式
	isnode	判断结点是否存在
	istnode	判断结点是否是终结点并返回排列值
	nodeasc	计算上溯结点
	nodedesc	计算下溯结点(子结点)
	nodepar	寻找父结点
	ntnode	求终结点的个数
	nodesplt	分割(分解)结点
	treedpth	求树的深度
	treeord	求树结构的叉数
	wdatamgr	管理数据结构
	wtreemgr	管理树结构

续表

■ 小波分析中的数据 I/O 函数		
	load	读取数据
	save	将 MATLAB 工作环境中的数据存盘
	fopen	打开一个文件或获取一个文件的信息
	fclose	关闭一个或多个已打开的文件
	fprintf	将数据以一定的格式写到文件中
	fwrite	将数据按照二进制格式写到文件中
	fscanf	从文件中按指定的格式读取数据
	fread	从二进制文件中读取数据
	ftell	获取文件指针的位置
	fseek	设置文件的位置指针
	ferror	获取文件 I/O 错误信息
	imread	从图像文件中读取图像数据(二维数据 I/O)
	imwrite	将图像数据按照一定的图像格式写入文件
	imfinfo	返回一个图像文件的信息

参 考 文 献

1 楼顺天、李博菡. 基于MATLAB的系统分析与设计(信号处理). 西安：西安电子科技大学出版社，1998.9

2 楼顺天、于卫. 基于MATLAB的系统分析与设计(控制系统). 西安：西安电子科技大学出版社，1998.9

3 楼顺天、施阳. 基于MATLAB的系统分析与设计(神经网络). 西安：西安电子科技大学出版社，1998.9

4 秦前清、杨宗凯. 实用小波分析. 西安：西安电子科技大学出版社，1998.1

5 张贤达、保铮著. 非平稳信号分析与处理. 北京：国防工业出版社，1998.9

6 宋建社. 小波分析及其应用例选. 北京：现代出版社，1998.5

7 刘贵忠、邸双亮. 小波分析及其应用. 西安：西安电子科技大学出版社，1997.3

8 程正兴. 小波分析算法与应用. 西安：西安交通大学出版社，1998.5

9 周舒梅. 动态信号分析和仪器. 北京：机械工业出版社，1990.10

10 L.科恩著. 时一频分析:理论与应用. 白居宪译. 西安：西安交通大学出版社，1998.3

11 杨福生著. 小波变换的工程分析与应用. 北京：科学出版社，1999.2

12 Kenneth. R. Castleman 著. 数字图像处理. 朱志刚等译. 北京：电子工业出版社，1998.9

13 Daubechies (1992). Ten lectures on wavelets, CBMS—NSF conference series in applied mathematics, SIAM ED

14 S. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. IEEE Pattern Anal. And Mathchine Intell., 1989, vol. 11, no. 7

15 胡昌华，许化龙. 控制系统故障诊断与容错控制的分析和设计，北京：国防工业出版社，1999.11